Applying Spatio-Temporal Analysis to Angle-Distance Views for Detection of Relevant Events

Georgii Khachaturov[1], José Antonio Espinosa de los Monteros Lizárrga[2], Josué Figueroa González[3]

Universidad Autónoma Metropolitana- Azcapotzalco, Av. San Pablo 180, Reynosa Tamaulipas, México, D.F., 02200, D.F., México.
[1]xgeorge@correo.azc.uam.mx, [2]jaespinosa777@gmail.com,
[3]jfgo@correo.azc.uam.mx

Abstract. This work deals with a virtual robot that operates in a 2D-environment populated with obstacles. Robot is supplied with a sensor that scans an angular field of view and produces an angle-distance plot. Regarding the flow of such plots as input, this work presents a processing scheme for detection of relevant events. Those, in turn, should serve for the non-Cartesian robot navigation by switching local maps in an atlas that would represent geography of the robot environment. The first step of the processing scheme transforms each plot to a string of singular points constructed as discontinuities of the first and second derivatives. On the second step, two such strings are subjected to comparison by a technique that employs the distance of Levenshtein; the distinctions between the strings lead to detection of relevant events. Special stress is put here on the algorithm of the first step and optimization of its parameters.

Keywords: non-Cartesian navigation; processing range images; map switching, Levenshtein distance.

1 Introduction

We begin from a description of how the particular problem treated in this paper is related to a wider research: the one of non-Cartesian robot navigation.

In the scope of a research program dedicated to fundamental issues of non-Cartesian navigation, we deal with a virtual robot that acts in a bi-dimensional environment populated with obstacles. Fig. 1 shows an instance of such environment; it is a supervisor's view of a scene; it shows the robot as a dark triangle at the bottom. The 'supervisor' refers to a user who controls the robot.

If the robot would be supplied with a color camera, it would see the same scene as Fig. 2 shows. However, it is assumed that the robot sensor is of a different kind: it scans all directions in a field of view and measures a distance to the first obstacle along each direction. Finally, each distance is subjected to a functional transformation so that the final view obtained by the robot for the same scene is the one shown in Fig. 3.

Note that calling below a plot like the one in Fig.3 as *'angle-distance plot'* should not lead to confusion in spite of the fact that it is not a crude, but transformed distance.

A single supervisor's view, Fig. 1, conveys full information about geography of the environment. But robot is unable to see it and its goal is to reach an equivalent understanding of the geography, but by means of a set of views similar to that in Fig. 3.

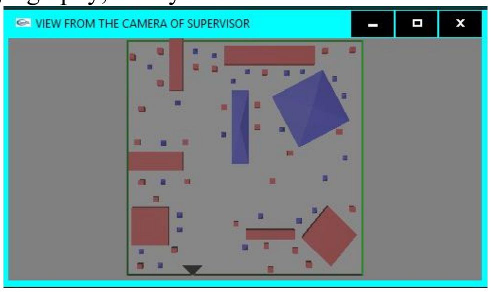

Fig. 1. An example of supervisor view of a robot environment.

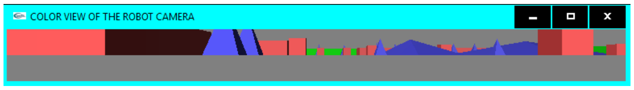

Fig. 2. The robot view of the same environment; the robot is shown as a dark triangle.

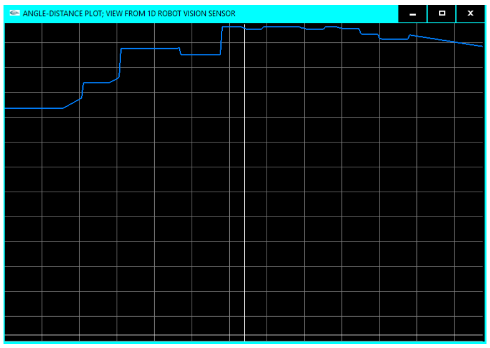

Fig. 3. The same robot view represented by an 'angle-distance plot'.

The process of learning geography is based on the moving robot along various routes in the scene combined with interpretation of the obtained 'angle-distance' views. It starts from an initial robot position in the scene, whereas the whole robot trajectory during learning must be continuous.

A crucial issue for this process is detection of 'relevant events', or in other words, those changes in the input flow generated by the robot sensors, which can be used as milestones for navigation. This work presents an approach to this issue.

The paper is organized as follows: Section 2 offers a review of some previous related works and the concepts treated in this paper.

Section 3 contains a development for detection of the relevant events.

Section 4 presents some specific issues developing furthermore the draft of Section 3, namely it deals with the extraction of singular points. Also it describes an optimization of parameters of the extraction algorithm.

Section 5 contains experimental results related to optimization of Section 4.
Section 6 contains a conclusion.

2 Concepts and Related Works

This Section presents some concepts and related works needed for understanding the current work.

2.1 The Levenshtein Distance

The Levenshtein distance [1] is a measure of the difference between two sequences of information, commonly applied to strings and it can be described as the minimum numbers of operations over single letters needed to convert or change one word into another. The process for obtaining this measure was developed by Vladimir Levenshtein in 1965. The definition of the Levenshtein distance between two strings *a* and *b* follows:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) \quad for\ i = 0 || j = 0;\ otherwise: \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)}, \end{cases} \end{cases} \quad \mathbf{(1)}$$

where $1_{(a_i \neq b_j)}$ is a function equal to 0 when $a_i = b_j$ and to 1 otherwise. Here *i* and $j \geq 0$ run indexes of symbols in respective strings. So this formula determines recursively all elements of a rectangle matrix. The Levenshtein distance is mainly used in detecting plagiarism in text or codes [2] and recently in dialect analysis [3], some uses of this technique related with image processing can be found in [4]. Here it is applied to the strings constructed as a coded form of the angle-distance plot.

2.2 Non-Cartesian navigation

Navigation for humans and animals does not assume pointing the goal as a point in a Cartesian space. The attempts to understand how the human brain tackles the navigation tasks have been undertaken in numerous works, during several decades mainly in psychology. The first mathematically strict model of non-Cartesian navigation was proposed by Khachaturov, [5]. Other works on this topic are [6], [7], [8].

The model presented in [5] (so called GT-model) is formed by two graphs, one of which represents geographic structure of a corresponding environment, meanwhile another is responsible for representation of robot knowledge of the kind "behavioral cause - sensorial consequence". It was shown that a navigational problem is reduced to a kind of the shortest path problem, so called 'extended shortest path problem', which, in turn, can be solved by a dynamic programming algorithm with a relatively low computational complexity.

2.3 Simultaneous Localization and Map Building Problem

The main goal of the research mentioned in the beginning of Section 1.1 focuses in developing techniques that allow a robot to navigate in its environment. Other works treat a similar problem called Simultaneous Localization and Map Building Problem (SLAM) [9]. Main of those works are related to creation of maps using specifics of the environment where a robot navigates, [10], [11].

3 A Draft Scheme for Extraction of Relevant Events

Here the logic of detection of relevant events is explained and justified. In the end it contains a concise description of the main idea of the processing.

3.1 Geography of the lowest level

In spite of the advantages of the GT-model mentioned in section 2.2, no progress in its practical implementation can be found in literature since the publication of [5]. The main obstacle for that consists in the necessity of creation specific tools for automatic learning the lowest, non-verbalizable, level of geography.

To clarify this issue, consider any geographic item that has an explicit name like Guadalajara, Broadway, Europe, etc. It is technically clear how to represent such an item in the geographic graph of GT-model. In contrast to that, each item of the lowest geographic level should be represented by a map that does not have neither distinctive shapes, nor an attributed name.

Moreover, in contrast to the verbalizable levels of the geographic graph of the GT-model, the lowest-level geography depends essentially on the sensor system. Note that the higher levels can be the same, say, for a robot, a blind person, as well as for a human who does not suffer any illness of sight. On the contrary, the lowest level geography for a robot depends on its sensors: for a blind person – on the stick which he/she uses for exploration of the environment, and for a normal person – on his/her vision.

This is why a special technique for learning geography of the lowest level should be developed and studied.

3.2 Sensor system for learning geography of the virtual robot

For understanding the main issues related to the just mentioned technique, a research was started at UAM-Azcapotzalco with the goal to develop a pilot system that would reach the functionality explained in section 1. In the scope of this project, a special attention is focused on the sensor input and its role for learning geography.

The geographic database to be constructed can be regarded as an atlas formed by maps. Robot should permanently resolve a self-localization task, which means to point out a map (the actual map) to which the current robot state belongs. Since the robot moves, the actual map sometimes must be switched to a neighbor map. In this context, the main task for processing the sensor output is to detect situations that should trigger a switch of a current actual map. We call such a situation as a *relevant event*.

The primary robot sensor is chosen for the pilot system as it was presented in section 1. This choice can be justified by several reasons:

While using a virtual reality platform, for example OpenGL, it is easy to generate a complex scene and extract any kind of information related to the scene for different kinds of virtual camera. Note that Figs. 1 – 3 are constructed just in this way for one and the same scene.

Processing one-dimensional images is obviously simpler than those two-dimensional. Furthermore, we believe that detection of relevant events by processing the angle-distance plots, Fig. 3, is even simpler than for the color images, Fig. 2.

We believe that integration of the components in the scope of the pilot project mentioned above is the primary goal. This is why the simpler components are more preferable on the first stage. More complex sensors should be considered on the next stages.

3.3 Main idea for detection of relevant events

We believe that the visual events specifically important for navigation are related to the facts of appearance/disappearance of objects in sight and to a qualitative change in appearance of an object. So they should be primarily detected and then used for indexing the maps of a geographic database.

On the other hand, note that while scanning an angle-distance plot, as the beginning as well as the end of an object in the robot sight are strongly correlated with discontinuity of the first derivative of the plot; note then that a discontinuity of the second derivative of the plot has a strong correlation with an angle of shape of an object in sight. This can be easily seen in Figs. 1 - 3.

This observation suggests us to perform a transformation of each angle-distance plot into a string formed by singular points constructed as discontinuities of the first and the second derivatives of the plot. This transformation is the first step in detection of relevant events.

The second processing step should perform a temporal analysis of the flow of such strings. The idea to use for this step a technique based on the Levenshtein distance suggests itself: when two strings of the singular points have been represented in an alphabet, one can find their editorial prescription. If the analysis shows that such two strings generated for some close moments completely match each other, it is naturally to claim that no relevant event occurred; otherwise, a further analysis of the vector of editorial prescription should follow to classify possible occurrence of a relevant event.

There are two issues which cannot be ignored in a further development of the described processing scheme. The former is related to the errors in classifying a singular point; it turns out to be that such errors cannot be avoided completely; however the algorithm of extraction of singular points can be optimized to reduce probability of the errors. The second one follows from the fact that the alphabet for the singular points is rather small which makes the errors of editorial prescription rather probable; to reduce probability of the errors, one need to involve in comparison of strings not only the symbol names, but some additional attributes of symbols (i.e., of singular points).

4. Detection of singular points

This section deals with a further development of the first processing step of section 3.3 and its optimization.

4.1 Specifics of angle-distance plot

An Angle-Distance Plots (ADP), like the one in Fig. 3, comes into our model from z-buffer [12] of a graphical API developed by OpenGL. As it was mentioned above, it represents not crude, but transformed distances. If for instance, an orthographic projection is used, then value z_b of z-buffer is produced from a real distance z as:

$$z_b = 2 * \frac{z-near}{far-near} - 1, \qquad \textbf{(2)}$$

where *far* and *near* are, respectively, maximum and minimum of allowed distances. This formula performs a compression of far distances into z-buffer to the values less than, but close to *far*. As result, the probability of confusion between an angle SP and a step SP grows for the far distances. In fact, it is impossible to avoid completely this kind of errors. However, their probability can be reduced to a reasonably low limit in the scope of a certain algorithm by means of optimization of its parameters.

In the rest of this section we describe our algorithm and how it has been optimized.

4.2 Extraction of a string of SPs from ADP

Beginning from the lowest argument value of an ADP, in parallel two processes are started: one of them tries to find a step-SP closest to the start argument, whereas another – a closest angle-SP. The former process seeks the closest local maximum of absolute value of the first derivative of ADP, whereas the latter – of the second one.

If SPs of both kinds are successfully found that lay on a long distance from each other, then that of them is chosen which is closer to the starting point. Otherwise, when the distance between them is short, an additional test is applied to select a better option.

Namely, we verify the hypothesis that attribution of SP as an angle is true. If so, then there should exist two non-vertical line segments in the ADP that meet each other closely to the position of the detected angle. Using a fragment of ADP close to this position, the best point is estimated where the left-hand line segment would meet the right-hand one. Finally, the hypothesis is accepted if the estimated meeting point stays quite close to the position of SP, otherwise it is rejected to give place to the step-SP.

The above process is repeated in a loop, starting from the last found SP. It follows up to exhaustion of arguments in the ADP.

4.3 Optimizing the detection of SP

The process described in the previous section depends mainly on the three thresholds: **lim_1st_der** responsible for rejection of a step-SP candidate with too small absolute value of the first derivative of ADP; **lim_2nd_der** responsible for rejection of an angle-

SP candidate with too small absolute value of the second derivative of ADP; **STEP_ANGLE_TOLERANCE** responsible to separate the 'long distances' from the 'short' ones in the meaning of the second paragraph of section 4.2.

To minimize the probability of wrong detection of SP, firstly, a generator of ADPs was developed. It generates a random ADP with a single SP in the middle of the plot; the kind of ADP was either a step, or an angle. All parameters of the left-hand and right-hand sides of the plot – vertical levels, derivatives, etc. – are also generated at random.

Then, in a loop, the output of this generator was sent to the processing of section 4.2 to measure the statistics of attribution of SPs to a certain kind.

Applying this technique, it was discovered that the thresholds chosen intuitively lead to the probability of errors about 8.0%. Then computation of a histogram shown that most of the errors are detected when the level of the left-hand side plot segment is quite close to that of the right-hand side. Based on these results, a process of optimization of the thresholds was performed. The thresholds were sought in some reasonable intervals around the initial values. The histograms helped us to modify parameters of the generator so that it would yield only those ADP for which a posteriori probability of errors is non-zero, which reduced drastically the processing time for optimization.

In result, for the optimized thresholds, the error of attribution of SPs to a certain kind has reached probability around 0.2 – 0.3%. See Section 5 for some details.

5. Experimental Results

A very short extract of our experimental results is presented in Table 1 and Fig.4.

Experiment #	**Error (%) by 10,000 Iterations per experiment**	**STEP_ ANGLE_ TOLERANCE**	**lim_1st_der**	**lim_2nd_der**
27	0.26	-0.75	0.0195	0.0031
28	0.26	-0.75	0.0098	0.0031
29	0.36	-3	0.0049	0.0031
30	0.38	-3	0.0391	0.0031
31	0.23	-0.75	0.0049	0.0024

Table 1. Errors of attribution of SP to a kind for a series of the best experiments.

6. Conclusions

A novel processing scheme for detection of relevant events from a flow of angle-distance views is developed. An algorithm for extraction of singular points from an angle-distance view generated by a virtual sensor is presented and optimized. After optimization, the probability of error of the algorithm was reduced from 8% to 0.2-0.3%. The presented technique is already employed in a research for non-Cartesian navigation. As well, it can be applied to the range images produced by a real sensor.

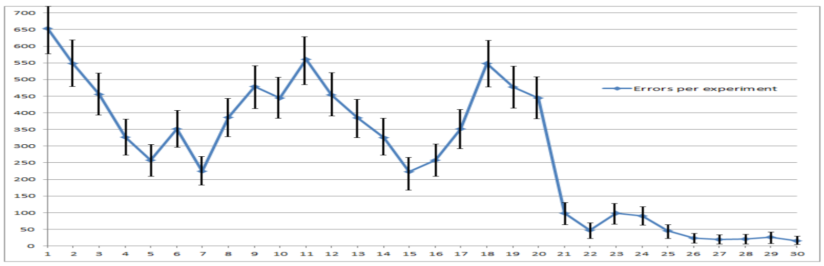

Fig 4. Number of errors in a wider series of experiments; vertical lines show 99.7%-confidence intervals (that is, intervals of 3σ-range); number of errors in the 1st experiment is 8% of sample.

Acknowledgments. The Fig. 1—3 were generated by software developed by Juan Manuel Martínez Hernándes, [13], for his graduation in 2010 from UAM-Azcapotzalco. UAM-Azc. supports this work in the scope of project "Visión Cognitiva Artificial".

References

[1] Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4), 845-848.

[2]. Su, Z., Ahn, B. R., Eom, K. Y., Kang, M. K., Kim, J. P., & Kim, M. K. (2008, June). Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm. In Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on (pp. 569-569). IEEE.

[3]. Heeringa, W. J. (2004). Measuring dialect pronunciation differences using Levenshtein distance (Doctoral dissertation, University Library Groningen][Host]).

[4]. Schimke, S., Vielhauer, C., & Dittmann, J. (2004, August). Using adapted levenshtein distance for on-line signature authentication. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on (Vol. 2, pp. 931-934). IEEE.

[5] Khachaturov, G. (2002). An approach to trip-and route-planning problems. *Cybernetics and Systems*, 33(1), 43-67.

[6] Gomi, T. (1996). Non-cartesian robotics. *Robotics and Autonomous Systems*, 18(1), 169-184.

[7] Gomi, T. (1997). Aspects of non-cartesian robotics. *Artificial Life and Robotics*, 1(2), 95-103.

[8] Vukobratović, M. (1997). How to control robots interacting with dynamic environment. *Journal of Intelligent and Robotic Systems*, 19(2), 119-152.

[9] Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on robotics and automation, 17(3), 229-241.

[10] Davison, A. J., & Murray, D. W. (2002). Simultaneous localization and map-building using active vision. *IEEE transactions on pattern analysis and machine intelligence*, 24(7), 865-880.

[11] Guivant, J., Nebot, E., & Baiker, S. (2000). Autonomous navigation and map building using laser range sensors in outdoor applications. Journal of robotic systems, 17(10), 565-583.

[12] Joy, K. I. (1996). The depth-buffer visible surface algorithm. On-Line Computer Graphics Notes, Visualization and Graphics Research Group, Department of Computer Science, University of California-Davis.

[13] Martínez Hernández, J. M. (2010). Desarrollo de un robot virtual basado en un modelo de navegación no Cartesiana y su entorno de desplazamiento. *Proyecto Terminal de la Carrera de Ingeniería en Computación*, UAM-Azcapotzalco.