View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Universiti Teknologi Malaysia Institutional Repository

Jurnal Teknologi

Full Paper

SHORTEST PATH PLANNING FOR SINGLE MANIPULATOR IN 2D ENVIRONMENT OF DEFORMABLE OBJECTS

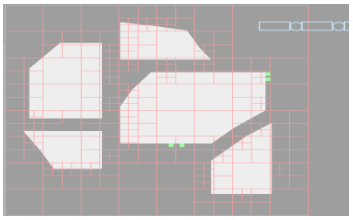
Fakhrul Syakirin Omar*, Md. Nazrul Islam, Habibollah Haron

Department of Computer Science Faculty of Computing, Universiti Teknologi Malaysia, 81310, Johor, Malaysia

Article history
Received
3 December 2013
Received in revised form
2 July 2014
Accepted
25 November 2014

*Corresponding author
fsyakirin2@live.utm.my

Graphical abstract



Abstract

A heuristic algorithm to perform path planning for single manipulator in 2D environment containing deformable objects is presented. The environment is partitioned into a quadtree hierarchy for both sampling and space navigation use before combination of artificial potential field and heuristic reasoning are applied iteratively to generate feasible path for the manipulator. The algorithm specifically targets for the shortest path without damaging any objects due to deep collision depth between manipulator link and object. Resulting path is in turn to be used in generating micro-instruction controlling the manipulator. Implementation results show feasibility to solve problems involving simple object and manipulator configuration.

Keywords: Path planning, single manipulator, deformable object, robotic surgery, CUDA

© 2015 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

Motion planning problem is a topic in robotics field dealing with automation of path searching throughout the map or environment from start to goal position or configuration along with providing complete instructions to guide the mobile robot or agent moving along the path. Path choosing criteria usually dependent on multiple factors to be considered correct or feasible, for example the path being collision-free, in all configurations the objects are not damaged, or is the shortest path possible. It can be found in wide range of problems such as independent agent navigation and task optimization, in addition to robotic surgery as medical application and virtual characters in games.

Minimally invasive robotic surgery (MIRS) is an application of robotic equipment in medical treatment which aims to decrease patient recovery time after the operation due to trauma from incision by keeping the cut size small compared to regular operation. This kind of operation is also known as keyhole surgery. Equally, manipulating the tools is harder as its degree of freedom decreases as well as limiting the patient insides view and tactile sensation from the surgeon. Thus, having information in advance about the surrounding area is advantageous which can be obtained via X-ray and other imaging technologies for visual confirmation, in addition to planning the tool trajectory beforehand to avoid risk puncturing weak organs or searching for easy path to handle the tool for the surgeon.

This paper proposes a method to search working path that can handle patient's internal body environment which contains deformable objects that has limitation to their surface pressure intake; it will be damaged if the limit is breached. It explores the environment map via space sampling paired with a quadtree for easy search and navigation, then testing and correcting the manipulator configuration at each step. The algorithm supports only a single objective namely finding shortest distance possible path found. An overview about deformable object simulation is provided in Section 2 including review for existing algorithm for path planning algorithms in environment with such objects. Details of the algorithm are described in Section 3 including its development and steps involved, including explanation on implementation and experiment flow. Analysis and discussion for results obtained from the experiment is placed in Section 4, and finally Section 5 presents the conclusion and listing of possible improvements and future work.

2.0 BACKGROUND INFORMATION ON RESEARCH

Approach to solve path planning problems has been divided into methods used to explore the working environment. The most popular is sampling-based method [1] which tries to check an area of map region which gets more specific as time progresses. Probabilistic roadmap (PRM) [2-3] and rapidly exploring random tree (RRT) are contained in this class. Main drawback for algorithm of this class, due to its probabilistic nature, is inability to declare no solution exists if there is none, though the probability to produce a solution will be close to 1 over time. Thus, this class has probabilistic completeness as opposed to deterministic completeness. Other available methods are artificial potential field (APF) [4] and working with configuration space representing original environment [5-6].

Motion planning has been introduced in medical field for various situations from in providing a reference path for surgery [7-8] to suturing and needle steering automation or guide [9-11]. Common procedure to set up the virtual working environment is by having internal image of the patient taken as working map then setting object properties such as firmness and limiting factor to match the actual organ. The objective for searching a path ranges from having a known safe and optimal route to follow to include detailed controlling instruction to be taken during manipulating surgery tool. Taking this further, a fully automated robotic surgery system could be formed with little or no human intervention needed in order to perform the operation.

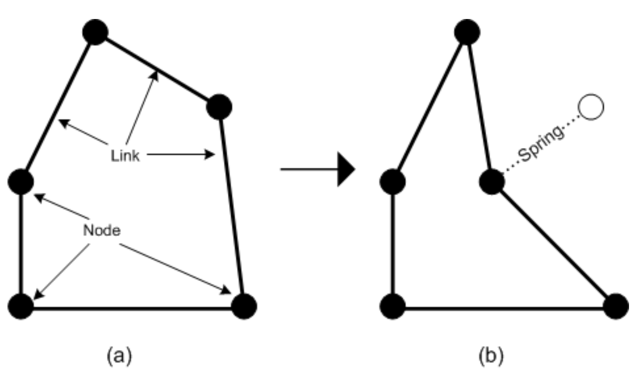


Figure 1 Structure of deformable object; (a) Original object shape; (b) Hidden spring shown connecting each node to their original position

Attempting to build such complete surgery automation, this paper concerns with navigating the manipulator through the environment while emphasizing objective that related to surgery that is the shortest travel path to the target. The algorithm presented falls into sampling type, aiming to provide robust but simple method in order to enable scalability when large working environment area is involved.

3.0 PROPOSED MOTION PLANNING ALGORITHM

This section defines the algorithm, divided into 3 major parts: spatial map partitioning using quadtree, iterative tree leaf moving based on APF and heuristics, and correction chance from deformable object surface collision responses. Additionally, implementation and experimentation flow of the algorithm is explained at the end of the section.

3.1 Deformable Object and Manipulator Properties

The The environment for algorithm testing consists of 2 types of object: generic deformable solids and a single multiple-link manipulator. Each deformable object is formed by a polygon shaped with connected nodes with its deforming behavior is defined by a single spring link between the nodes as shown in Figure 1 above. Its main purpose is to provide penalty feedback due to deformation during collision event with manipulator by responding dynamically to node positional displacement. As for the manipulator such in Figure 2 below, rigid rectangles are connected with rigid circles at each end to create chain which can be either rotated at each link level or moved for the entire frame by moving its base.

The deformation occurs when any collision event between any object and manipulator link is detected. After object nodes involved with the event are identified, its position is modified to follow the shape of intruding link. Further deformation process taken place follows the pattern described in [12], which when completed provides feedback force acting at the link and state of the material whether any link has been overstressed.

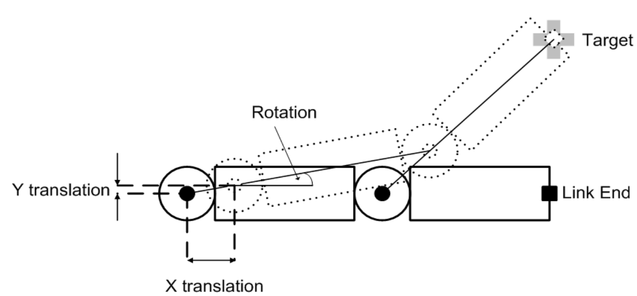


Figure 2 Structure of the manipulator used and its control properties, obtained during inverse kinematic operation when moving to next leaf

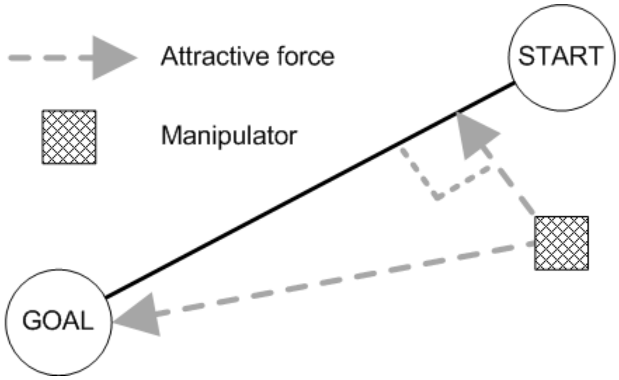


Figure 3 2 different attraction forces affecting the manipulator direction

3.2 Partitioning Environment using Quadtree

Utilizing the clue that object outline is defined by its surface nodes, space sampling by counting node population in each imaginary region is performed. If the count reached over the prescribed limit, the region is divided further until all child regions are below this limit. This behavior introduces adaptive region resolution for the end result due to the sampling region will be more detailed and smaller in size where the surface node concentration is higher, which is usually where the objects are touching. Such areas are of interest with its location is where the path is expected to be since pressure to the object is minimum as the manipulator will be penetrating slightly at the surface.

The partitioned region is then compiled into quadtree hierarchy with easy navigation in mind, where the tree leaves serve as the only standing points possible in the map for probe to move on. This data structure is chosen due to its simplicity to generate and matching the 2D environment with its 4-child leaf per parent cell. The leaf cell is defined as a region that has no further subregion in it, in contrast with a parent cell which is the opposite. The resulting route then will be starting at a leaf where the exact starting coordinate is in and ended at another.

3.3 APF- and Heuristic-based Iterative Tree Leaf Moving

There are two sources of attraction forces defined in the algorithm: along a straight line connecting starting and goal position and the goal itself. The former reflects the ideal shortest path possible to be followed by pulling the manipulator at right angle while the latter guides to its general direction. The combination of the force vectors are used to calculate next moving angle to take, that is, which neighboring leaf will be chosen to go next. The illustration of the process is done in Figure 3 above.

Since the tree hierarchy is a combination of parent and leaf cells, and the requirement that only leaf cells are accepted for mapping usage, some method could be useful to ease traversal of the tree. As such, a solution outline to this problem is provided in Figure 4. Basically, after the moving angle is known, the current cell boundary that corresponds to the direction is recorded and compared with every other cell boundary during the search. When there are multiple conflicting neighbors sharing the wall, possibly due to smaller cell dimension, its wall span is converted into angle range as the definitive indicator to be compared with the original moving angle.

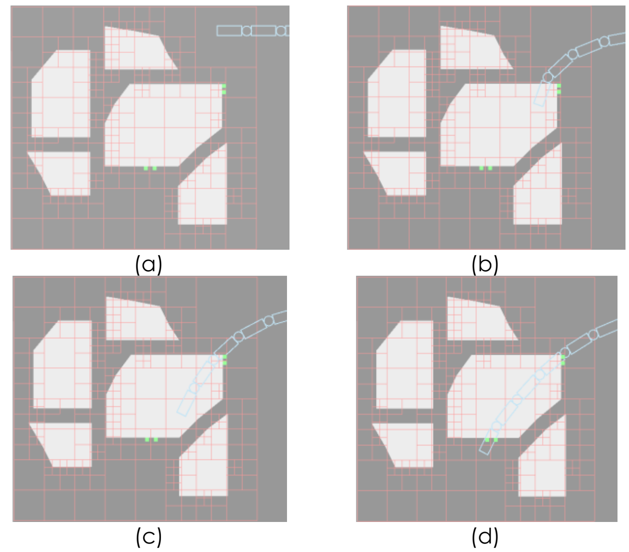


Figure 4 Solution outline

After the next leaf to move to is identified, the manipulator is then rearranged from current position for its end-effector to reach the centre of that cell. This paper applies straightforward inverse kinematics for the calculation sequentially from the manipulator end up to the base. This manipulator configuration change is done to test the feasibility of including the next leaf to the path, particularly existence of collision with any object and its degree of deformation. At this point also micro-instruction that controls the manipulator can be obtained and recorded to be used for actual manipulator device.

3.4 Path Correction by Object Surface Collision Response

Path feasibility factor includes restriction that at any step there are no overstressed object linkage that signs damage due to the manipulator. When such an event occurs, the algorithm tries to correct the leaf chosen by following the direction of feedback force from the object surface. Applying the same pattern as in previous Subsection C, current next leaf chosen in previous section acts as the base to find new neighboring leaf which in turn replaces the current next leaf thus avoids the damaging situation.

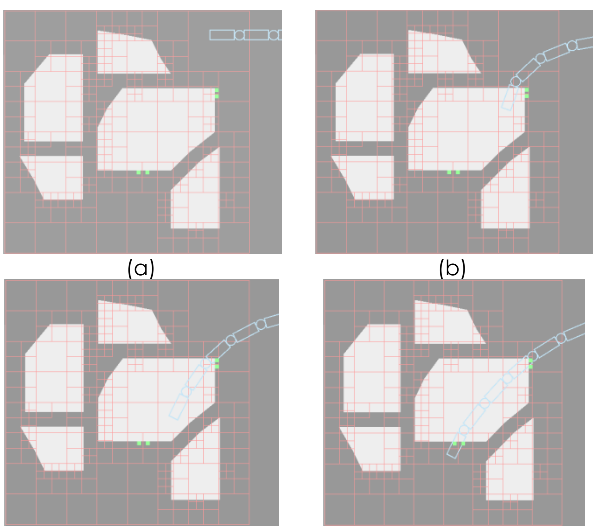
Further correction to the algorithm end result is to find any redundant leaf in found path, which is whether there is any path segment that can be reconnected by bypassing its intermediate leaves. In this stage, the manipulator is still needed to be moved along the path to test such possibility. Improvement per subpath technique can be traced to PRM method, which answers path queries for a long distance by planning a small segment of it recursively.

3.5 Implementation and Experimentation

The algorithm is implemented using CUDA C running on NVidia GT440 graphic card with AMD Athlon II 3GHz and 4GB RAM. The program is tested on different setup by varying object location and properties, and manipulator link count and length. The benchmark is to see algorithm ability to find any valid path and its criteria in terms of achieving separate path objectives: objects deformation degree and total distance covered.

4.0 EXPERIMENT RESULT AND ANALYSIS

In Figure 5, progresses of the algorithm in steps are shown with the manipulator moving towards the goal from its starting configuration. Even if there are collision event existed from the movement following the path as indicated from small green rectangles at the boundary of deformable objects marking object boundary nodes, as long the inflicted pressure does not damage the material, the path is accepted as feasible.



(a) (b) (c) (d)

Figure 5 Progress of the algorithm in (a) to (d) sequence. The manipulator (blue rectangle/circle) ignores deformable object (gray) in its path as long as it does not inflict too much pressure to damage it. Red lines shows leaves cells

There is no attempt to test the entire map for other possible paths as the accumulated APF is exclusive in a single direction. Thus, branching factor such as collision event with the object could be used as signal to choose next neighboring leaf at the exact opposite of the APF direction.

5.0 CONCLUSION AND FUTURE WORK

Based on the experimental result, the proposed algorithm shows the ability to find feasible paths in simple map setting. Using combination of 2 potential field sources: reference path and goal position, and heuristic adjustment, shortest path in term of manipulator travel distance can be found.

Various improvements to the algorithm and restriction can be done in order to handle larger environment diversity [13] and increase feasibility confidence which is listed here:

- Manipulator link rotation restricted to a small gap apart from other links, which signify manipulator state checking event at every step.
- The tree leaves are interconnected with ropes, a property proposed in [14] to help movement from a leaf to its neighbor.

Acknowledgement

This paper is sponsored by Zamalah student funding and Research University Grant project vote number 4D057 from Universiti Teknologi Malaysia and Malaysian Ministry of Higher Education.

References

[1] Carpin, S. 2006. Algorithmic Motion Planning: The Randomized Approach. General Theory of Information Transfer and Combinatorics. In Ahlswede, R., Bäumer, L., Cai, N., Aydinian, H., Blinovsky, V., Deppe, C., Mashurian, H. (eds.). Springer Berlin.

[2] Sánchez, A., R. Cuautle, R. Zapata, M. Osorio. 2006. A Reactive Lazy PRM Approach for Nonholonomic Motion Planning. Advances in Artificial Intelligence - IBERAMIA-SBIA 2006. In Sichman, J., Coelho, H., Rezende, S. (eds.). Springer Berlin.

[3] Jaillet, L., and T. Siméon. 2008. Path Deformation Roadmaps. Algorithmic Foundation of Robotics VII. In Akella, S., Amato, N., Huang, W., Mishra, B. (eds.). Springer Berlin.

[4] Mohamed E.F., K. El-Metwally and A.R. Hanafy. An Improved Tangent Bug Method Integrated with Artificial Potential Field for Multi-Robot Path Planning. 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA). Istanbul. 15-18 June 2011. 555-559.

[5] Igarashi, T. and M. Stilman. Homotopic Path Planning on Manifolds for Cabled Mobile Robots. Algorithmic Foundations of Robotics IX. In Hsu, D., Isler, V., Latombe, J.C., Lin, M. (eds.). Springer Berlin.

[6] Mahoney, A., J. Bross and D. Johnson. Deformable Robot Motion Planning in A Reduced-Dimension Configuration Space. 2010 IEEE International Conference on Robotics and Automation (ICRA). Anchorage, Alaska, USA. 3-8 May 2010. 5133-5138.

[7] Vaillant, M., C. Davatzikos, R. Taylor and R. Bryan. 1997. A Path-Planning Algorithm for Image-Guided Neurosurgery. CVRMed-MRCAS'97. In Troccaz, J., Grimson, E., Mösges, R. (eds.) Springer Berlin.

[8] Rilk, M., F.M. Wahl, K.W.G. Eichhorn, I. Wagner and F. Bootz. Path Planning for Robot-Guided Endoscopes in

Deformable Environments. Advances in Robotics Research. In Kröger, T., Wahl, F.M. (eds.). Springer Berlin.

[9] Vancamberg, L., A. Sahbani, S. Muller and G. Morel. 2010. Needle Path Planning Method for Digital Breast Tomosynthesis Biopsy based on Probabilistic Techniques. Digital Mammography. In Martí, J., Oliver, A., Freixenet, J., Martí, R. (eds.). Springer Berlin.

[10] Alterovitz, R., K.Y. Goldberg, J. Pouliot and I.C. Hsu. 2009. Sensorless Motion Planning for Medical Needle Insertion in Deformable Tissues. *IEEE Transactions on Information Technology in Biomedicine*. 13(2): 217-225.

[11] DiMaio, S.P., and S.E. Salcudean. 2005. Needle Steering and Motion Planning in Soft Tissues. *IEEE Transactions on Biomedical Engineering.* 52(6): 965-974.

[12] Omar, F.S., M.N. Islam, and H. Haron. 2012. Heuristic Modeling of Deformable Object using Node-based Structure with Mass-Spring System. 2012 International Conference on Interactive Digital Media (ICIDM). 1(2): 436-440.

[13] Rahim, M.S.M, S.A.M. Isa, A. Rehman and T. Saba. 2013. Evaluation of Adaptive Subdivision Method on Mobile Device. *3D Research*. 4(2): 1-10.

[14] Havran, V., J. Bittner and J. Zára. 1998. Ray Tracing with Rope Trees. Proceedings of 13th Spring Conference on Computer Graphics Budmerice in Slovakia. 130-139.