

Rectangular Maze Construction by Combining Algorithms and Designed Graph Patterns

Dr. Wen-Shou Chou
Associate Professor
Department of Digital Media Design,
Ming Chuan University, Taiwan

Abstract—Rectangular digital mazes can be constructed by algorithms automatically. However, visual aesthetics such as the shapes and structures of mazes can't be created gracefully due to the mechanic characteristics of algorithms. This study proposed a method to combine the maze generation algorithm and the graph patterns designed by designers, in order to make mazes can be constructed not only automatically but also creatively.

Keywords—maze; maze games; computer games

I. INTRODUCTION

Mazes and labyrinths have been used as pure visual art, such as the architectural decoration, or been built as the walk-through puzzles in corn-fields. Digital mazes can be built as stand-alone games, or be applied to construct levels of computer games. Besides, the printed puzzles are still popular till now. Mazes can be constructed either automatically by using algorithms or manually by applying drawing tools [1]. With manual method, mazes can be designed more gracefully, but it is very time consuming. While applying algorithms, mazes can be quickly constructed, but due to the characteristics of algorithms, the constructed mazes appear as “mechanical” as compared to those mazes designed by designers manually (Fig. 1).

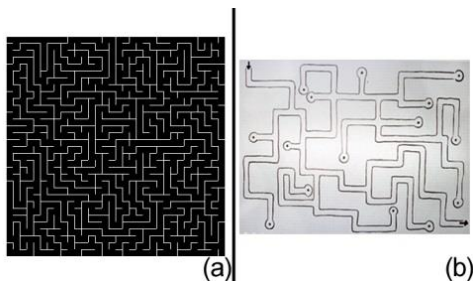


Figure 1. Maze constructed by algorithm (a) and designers (b).

Recent researches attempted to create image-resembling mazes [2, 3]. They created mazes algorithmically by referencing to the provided images. These mazes are less creative as compared to those designed manually. Furthermore, there may be no suitable images can be referred to while constructing the mazes. How to combine the automatic merits of algorithms and the creative characteristics of designers to build mazes seems to be a good design method. However, little research has been studied for this approach.

Labyrinth is a maze with single path that simply winds its way from start to finish with no branches. Unlike other general mazes, labyrinths are unsuitable for being used as printed puzzles, or be applied to digital games, since there are no ways to go wrong for players. While considering the construction of mazes for computer games, it is expected that the shape or structure of mazes can be adjusted dynamically in order to increase the challenge, to enhance the game entertainment, or to create the consecutive levels. Although current algorithms can produce a whole new maze for a new game level, it is not trivial to either dynamically change some part of the maze, or to control the whole shape or structure of mazes.

In this study, we present a simple way to combine the maze generation algorithm and the graph patterns designed by designers. Due to the modular feature of this method, it is easy to adjust the shape or structure of mazes dynamically.

II. MAZE BASICS

A rectangular maze can be viewed as having m cells in width and n cells in height. Each cell is consisted of up, down, left, and right four cell walls. If the wall between two neighboring cells is broken, these two cells are connected to form a path. A maze is also a puzzle, it essentially contains a starting cell, an end cell,

and the player is to find a solution path from the starting cell to the end cell. If there exists one and only one path between any two cells in a maze, the maze is a perfect maze. Otherwise, it is a non-perfect maze. Most of the mazes generated by using algorithms are all perfect mazes [4].

If we regard each cell as a node, the carved invisible walls as links, we can express a maze as a graph. Since there are no circles existing in a perfect maze, a perfect maze is basically a spanning tree [5]. The maze generation algorithms are therefore nothing more than algorithms for constructing random spanning trees. Some well-known algorithms include: (1) Depth-first Search (DFS) algorithm [6]. (2) Kruskal's algorithm [7]. (3) Hunt-and-Kill maze generation (HKMZ) algorithm [8]. By using the DFS algorithm as an example, this method uses the stack as data structure. Firstly, a cell is randomly selected. Then, one of unvisited neighbor cells is selected. The wall between them is deleted, move to this new cell, and push this cell onto stack. If all of the four neighbor cells are all already visited, pop out the cell from stack (backtracking the cell) until one of unvisited neighbor cells of the pop out cell is found, and continues the cell visiting process. When all of the cells are visited, the process ends and maze is constructed at the same time. Since the maze is perfect, there always exists a path between any two selected cells. Therefore, we can select any two cells of a maze puzzle to be the starting and ending cells, respectively.

The generating algorithms involve random mechanisms in the wall-breaking process. As in the DFS algorithm, the order of selecting unvisited neighbor cells plays an important role in forming the path structure of maze. For example, if the unvisited horizontal (left or right side) cells have higher priorities to be selected than the vertical (up or down side) cells in the wall-breaking process, the path structure of maze will become more horizontal. We can assign weights to cells or cell walls in order to control the path structure of maze [4]. This principle can be applied to control the appearance of mazes as described in the following section.

III. MAZE APPEARANCE

Previous studies about the maze appearance could be roughly classified into two categories: the maze art, and the image-resembling mazes. For maze art, Christopher Berg, the maze artist, he draws mazes based on monuments and ancient wonders [1]. In his works, the path routing pattern of maze could be viewed as being a “brush pattern” while “painted” these works.

Pedersen and Singh [9] considered the maze as bearing resemblance with point distribution patterns generated by stochastic sampling algorithm. They proposed a geometric attraction-repulsion model that can evolve a simple shape into an organic labyrinth. Their works are visually similar to the Traveling salesman art [10]. These works of maze art principally tried to generate labyrinth like patterns to mimic the shapes of some natural objects.

Xu & Kaplan [2] proposed an algorithm to construct vortex mazes. A vortex is a collection of passages that wind around each other in a spiral and converge on a central junction, wherein each spiral can be viewed as an alternative of labyrinth with the property of moving the dead end very far along the passage from the entrance. The spirals or vortex mazes are suitable to be used to fill in some ellipse or circular like regions. In the same year, they proposed a method to construct mazes resembling user-supplied images [11]. After designer manually partitioned the supplied image into a set of regions and assigned the style parameter for each region, the computer generated the filling-in maze patterns into these regions automatically, then, the walls between regions were broken to form a maze. In order to form maze textures in regions, they generalize the rectangular maze construction method by using two orthogonal streamlines [12] to form the grids. Their method can't preserve the salient structures in the source image essentially. Wan et al. [3] presented a reaction-diffusion simulator based on the cellular neural networks [13] to evolve image-resembling mazes from the source images. Their method could retain the interior structure in the images by using stripe patterns. However, user-specified guiding curves and post-processing methods should be given to turn the evolved stripe patterns into a maze.

Instead of using the contour information of supplied images to form mazes, Bosch & Herman [14] segmented the supplied image into K by K cells firstly. Then, randomly generated points for each cell (the number of points was proportional to the mean of gray levels in the cell), and applied the algorithm of solving traveling salesman problems (TSP) to find a path connecting these points. This work was similar to the Traveling salesman art proposed by them in the next year [10]. For applying the pixel information to generate image-resembling mazes, we can first extract out the image object from background to form a mask, which is used to constrain the area to form the whole shape of maze (Fig. 2(b)), then approximate the binary mask area with rectangular grids, and apply the DFS algorithm to generate maze. Wherein the mean of image gradient magnitude of each cell is used as the searching priority of the cell (greater mean with higher priority) in

the wall-breaking process of the DFS search algorithm. The image gradient magnitude (Fig. 2(c)) can be estimated by the edge detection filters such as the Sobel operator or the Prewitt filter [15]. As shown in Fig. 2(d), the path structure of maze is aligned to the shape texture of flower.

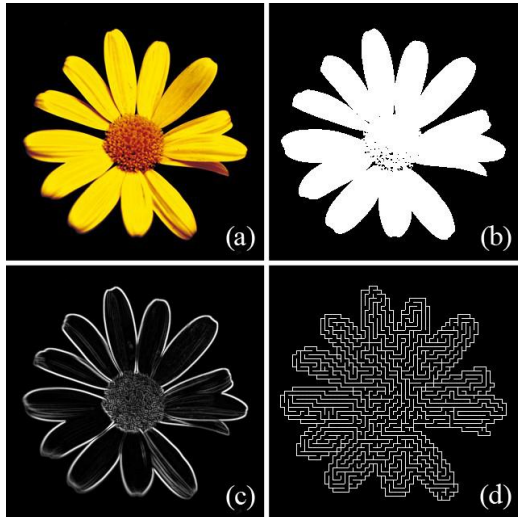


Figure 2. Maze constructed by applying image processing techniques: (a) source image (b) mask area (c) image gradient magnitude (d) the constructed maze (proposed by the author).

Unlike the approach of image-resembling mazes, the puzzle company Conceptis Limited [16] introduced a normally rectangular maze generated by algorithms, however, the cells that make up the solution path, when filled in, will reveal a hidden image. From the view point of puzzle games, maze art works are not suitable for being used as puzzle games, and the path structure of image-resembling mazes are still too “mechanical”, as compared to the free-form path structure designed by designers.

IV. THE PROPOSED MAZE CONSTRUCTION METHOD

We proposed that the whole maze is consisted of micro maze cells. Each micro cell can be classified into one of the five abstract categories (Fig. 3), wherein the figure, the black dots represent the entrance or exit of the micro maze cell. By using the B_0 cell as an example, it represents the micro maze cell can enter from the left and leave out to the right or vice versa. For the D_0 cell as another example, the maze cell can enter from one of the three sides, and leave out by one of the other two sides. The abstract type with index 0 in each category is denoted as the standard or canonical cell. And the other cells with index value greater than 0 can be obtained by rotating the canonical cell. In order to “soften” the path

structures of mazes as originally constructed by algorithms, the path details contained in micro maze cells are not specified and can be designed freely by designers. The system architecture of proposed method

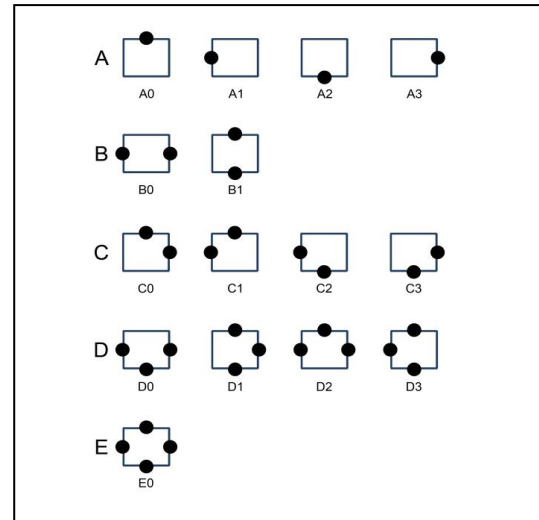


Figure 3. The abstract types and categories of micro maze cells

is shown in Fig. 4. Firstly the area of rectangular maze is segmented according to the size of micro maze cells. The maze construction algorithm (DFS algorithm was used in this study) is applied to get the random spanning tree. By referencing to the “code books” (the abstract types of cells shown in Fig. 3), each cell can be coded and we will get the coded spanning tree. Finally, the maze realization block decodes each coded cell by randomly selects and transforms the corresponding micro maze cell from designed graph pattern libraries to implement the whole maze.

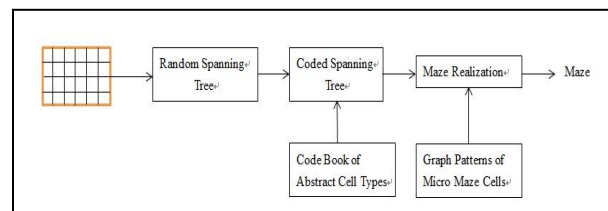


Figure 4. System architecture of the maze construction method.

The graph patterns of micro maze cells are designed by designers according to the specified abstract cell types. Some of the examples are given in Fig. 5 for illustrations. It should be noted that only the graph patterns of canonical type for each category are needed to be designed. Other type cells in the same

category can be implemented by rotating the canonical patterns in the maze realization processing block.

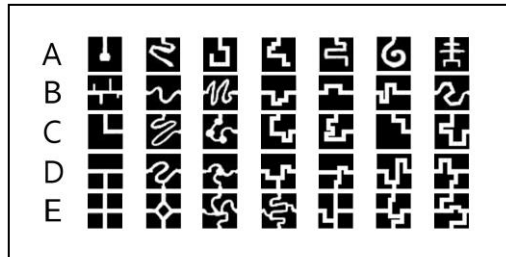


Figure 5. Examples of designed graph patterns.

Some of the constructed mazes with graph patterns designed by four designers are shown in Fig. 6. It can be seen obviously that the four mazes have distinct styles due to the micro maze patterns designed by different designers. The mazes in Fig. 6 (a) and (b) are perfect mazes, and the mazes in Fig. 6 (c) and (d) are not, since there exist loops in the designed micro cells. By mixing the graph patterns of micro maze cells in the maze generation process can produce a large variety of mazes. As examples shown in Fig. 7, Fig. 7(a) was constructed by mixing the two graph pattern sets as used in Fig. 6 (a) and (b), and Fig. 7 (b) used the pattern sets in Fig. 6 (a), (b), and (c).

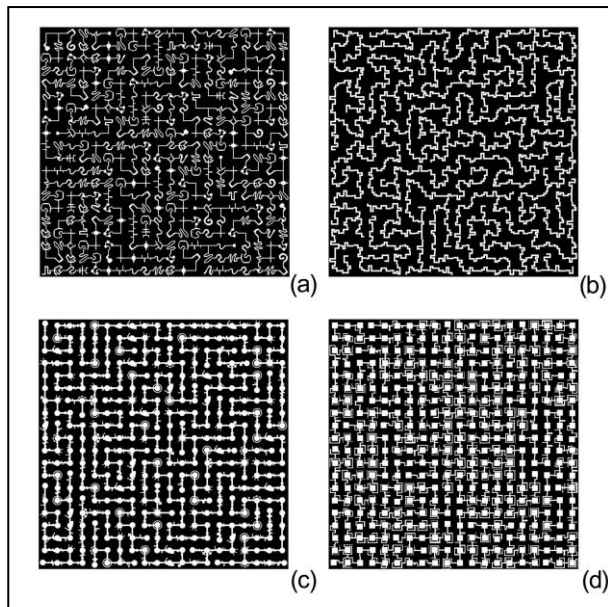


Figure 6. Examples of mazes constructed by the proposed method.

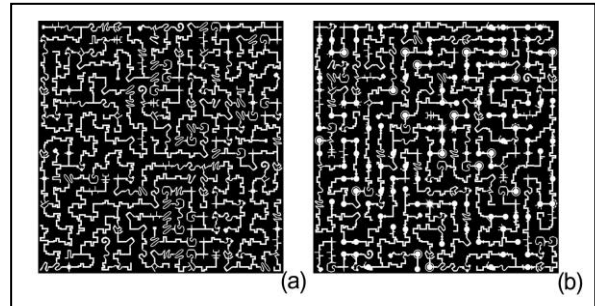


Figure 7. Examples of mazes constructed by mixing the graph patterns of designers: (a) mixing patterns of two designers (b) mixing patterns of three designers.

V. CONCLUSIONS

The creation of engaging mazes requires both information engineering and aesthetic considerations. In this study, the maze construction process can be divided into two parts: the design of graph patterns to be used as micro maze cells, and the algorithm for automatically selecting and transforming the micro maze cells to piece up the maze. This method can gracefully combine the artistic works of designers and the automatic feature of algorithms. Moreover, by mixing or changing the graph patterns of micro maze cells, the appearance of mazes can be dynamically adjusted, and a variety of mazes can be easily constructed.

ACKNOWLEDGMENT

This work was supported by the National Science Council of the Republic of China under contracts NSC 102-2410-H-130-028-MY2.

REFERENCES

- [1] C. Berg, *Amazing Art: Wonders of the Ancient World*, Harper Collins., 2001.
- [2] J. Xu, and C. S. Kaplan, "Vortex Maze Construction," *Journal of Mathematics and the Arts I*, vol. 1, no. 1, pp. 7-20, 2007.
- [3] L. Wan, X. Liu, T.-T. Wang, and C.-S. Leung, "Evolving Mazes from Images," *IEEE Transaction on Visualization and Computer Graphics*, vol. 16, no. 2, 287-297, 2010.
- [4] W. Pullen, "Think Labyrinth Maze Algorithms," Retrieved from <http://www.astrolog.org/labyrnth/algrithm.htm>, 2014.
- [5] B. Bollobas, *Modern Graph Theory*, Springer, 1998.
- [6] D. C. Kozen, *The Design and Analysis of Algorithms*, Springer, 1991, pp. 10-24.
- [7] T. H. Cormen, C. E. Leiserson, and R.L. Rivest, *Introduction to Algorithm*, MIT Press, 1992.
- [8] H.-L. Lee, C.-F. Lee, and L. -H. Chen, "A perfect maze based steganographic method," *The Journal of Systems and Software*, vol. 83, pp. 2528-2535, 2010.
- [9] H. K. Pedersen, and K. Singh, "Organic labyrinths and mazes," *Proceedings of the 4th International Symposium on*

- Non-Photorealistic Animation and Rendering, ACM, New York, USA, pp. 79-86, 2006.
- [10] C. S. Kaplan, and R. Bosch, "TSP Art," Proceedings of Bridges, Mathematical Connections in Art, Music, and Science, pp. 301-308, 2005.
 - [11] J. Xu, and C. S. Kaplan, "Image-Guided Maze Construction," ACM Transactions on Graphics, vol. 26, no. 3, Article 29, pp.1-9, 2007.
 - [12] B. Jobard, and W. Lefer, "Creating evenly-spaced streamlines of arbitrary density," Proceedings of the Eurographics Workshop in Boulogne-sur-Mer, France, Springer Verlag, pp. 43-56, 1997.
 - [13] L. O. Chua, A Paradigm for Complexity, World Scientific Publishing Company, 1998.
 - [14] R. Bosch, and A. Herman, "Continuous line drawings via the traveling salesman problem," Operations research letters, vol. 32, no. 4, pp. 303-320, 2004.
 - [15] A. McAndrew, Introduction to Digital Image Processing with MATLAB, Course Technology, U.S.A. 2004.
 - [16] CONCEPTIS LIMITED, "Conceptis puzzles." <http://www.conceptispuzzles.com>, 2014.

AUTHORS' PROFILE



Dr. Wen-Shou Chou is an Associate Professor in the Department of Digital Media Design, Ming Chuan University, Taiwan. He received the Ph.D. degree in Electrical Engineering from National Tsing Hua University, Taiwan in 1992. His current interests are in the area of interactive media design, game design, and new media art.