

Generation of Complex Picture Maze by Making Path along Pre-Positioned Image Dot Vertices

ABSTRACT

There are three interests in this study. First is to make an analogy between graph theory and image processing. Given a number of image items (dots) on a picture, this paper shows a way to make a path (closed path) connecting and traversing all of them but each only once, imitating to construct a path (closed path) on a plane by considering a dot as a vertex and a connecting line as an edge. This looks like constructing a Hamiltonian path on a plane and drawing a plane graph connecting all vertices with no crossing among the edges. Second is to construct the maze based on the obtained path with some detouring within each of image items (dots). This path could be very long and complex. Third is to apply this path making procedure to different pictures with different image shapes like symbols or English letters --- possibly generating a complex picture maze. Pictures or some meaningful message appear when the maze is solved. Algorithm and its proof are given with a number of successful experimental results.

Keywords: Maze, Picture maze, Graph theory, Path, Hamiltonian path, Plane graph, Image processing

1. INTRODUCTION

A maze in this study is a square maze organized in a computer array [1]. It consists of paths and walls. Path consists of a set of vertices and a set of edges like a graph. But more restraints are imposed on the maze. Vertex of the maze is placed one in every two cells vertically as well as horizontally in the array. If two adjacent vertices are connected, then there is an edge between the two and all the vertices in the array are connected but no cycle is allowed. So the maze is a spanning tree on the array. If two vertices on the tree is designated as the start and the goal, then the path between the two is the solution.

A maze constructed in this study is based on the maze structure above. But it is generated based on an original binary picture. Employing techniques of image processing, graph theoretical concepts, and following the constraints of the maze structure, a kind of complex maze (picture maze) is generated. Showing a part of the experimental results would rather work better to make the introduction of this study and should be more persuasive.

Fig. 1 is the original picture from which a maze would be constructed. Through some processes, a maze such as in Fig. 2 can be obtained. However, it is the constructed maze with the solution path (red) painted on it. If the dots have some shapes, then some pictures could appear accordingly.

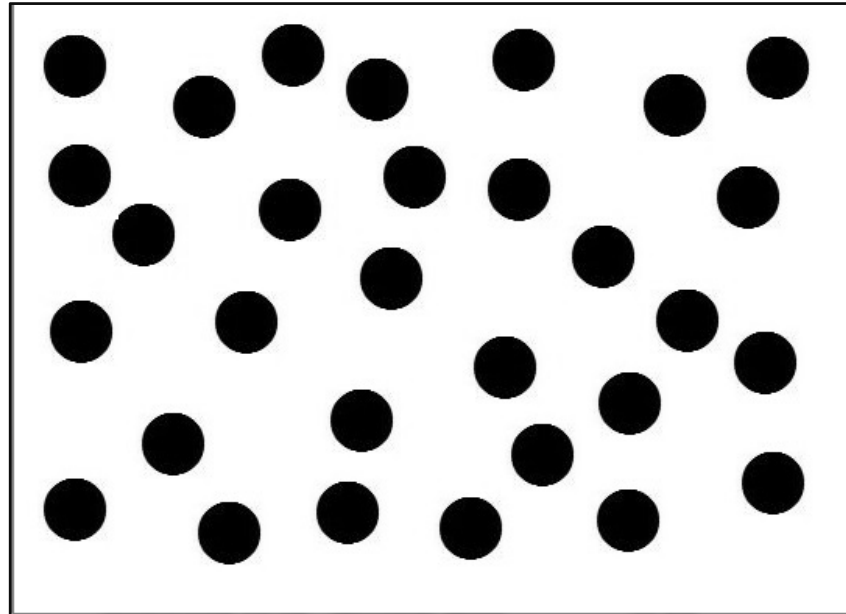


Fig. 1 Original picture of image dots

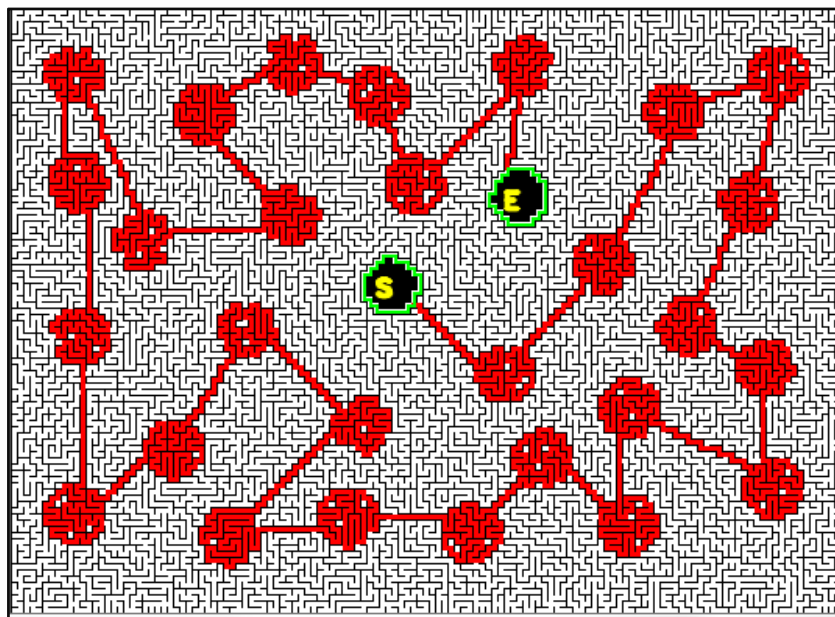


Fig. 2 Constructed maze of image dots with solution path

2. LITERATURE

Maze generation by computing on square grids has been around for some years. The article [2] showed various algorithms to generate a square grid based maze, usually used, relating with graph theory. Hosokawa [1] presented a number of maze generation algorithms with the relation between path, wall, and coordinates. Xu and Kaplan [3] illustrated complex mazes emphasizing aesthetics for the maze. They also introduced a picture maze [4] called 'Maze-a-pix' by illustrating a sample picture maze, which is provided by Conceptis, Ltd. [5]. However, the method of the maze generation was not given. The internet site of the company shows various picture mazes and states a history of computer generated picture maze. Okamoto and Uehara [6] demonstrated an automatic generation algorithm for a picture maze by constructing a Hamiltonian path which fills up the picture region. However, the path is generated on the 2-by-2 extended array. The maze is so constructed that the player obtains the solution path by filling up all the path cells on the maze and by going through the path. Hamada [7, 8] improved the method [6] by enabling to place the start and goal at different positions. Kurokawa developed a more efficient algorithm of construction of Hamiltonian path with a rigorous proof to generate a picture maze and introduced a generalized idea for maze path construction [9]. Ikeda and Hashimoto [10] illustrated a picture maze generation method by using simulated annealing. Kurokawa, Mori, and Mizuno [11-13] took a different approach and demonstrated the method to generate a picture maze by making use of the picture contours. In the method, all the picture contours are extracted and are incorporated into the maze solution path. It is a very flexible method and can be applicable to a wide range of binary pictures. F. J. Wong and S. Takahashi [14] constructed a hybrid picture maze from two photograph images. This maze is not based on the square grid type mazes but on the special grids computed from the image. Two photo pictures are overlapped in the maze. The maze player depicts an outline of one picture on the other picture background.

Although not about the picture maze construction, the study [15] is the one that makes a typical sample for relating the graph theory and the image processing. It extracts a Euler Circuit from one stroke drawing of a closed path by making use of the contours of the drawing and generates the maze solution path by applying graph theory including by-partitioning to the set of the contours. The present study itself also is a try of making an analogy between graph theory and image processing---realizing a Hamiltonian-like path by connecting image items on an image plane.

3. PRECEDING STUDIES

This study is about one of picture maze generation methods. Graph concepts were incorporated in image processing and maze construction. There are a number of preceding studies which have made some influences to this study. First is the one that made a picture maze by merging image contours [11-13]. In the study, image contour cycle extraction and contour merging played important roles. The second is to generate Hamiltonian-like path within an image item so that some image or picture appears when the maze is solved [9]. The third is Eulerian circuit, organized from a closed path of one stroke drawing [15], features of which brought a hint for doing the research on the path construction in this study.

4. CONCEPTS OF THIS STUDY

The purpose of this study is three-fold. First is to examine if it is possible to construct a closed path which traverses all the image dots, usually of large number (Fig.1), each once, which are randomly scattered on a plane. This is to make an analogy between image processing on image plane and graph theory and generate a closed path-like structure on an image array. An image dot (Fig. 1) is considered as a vertex of a graph and a connecting line between dots

as an edge. This problem is also considered as drawing a closed path (a plane graph) traversing all the vertices on the plane with no edge intersection.

As stated, second is to construct the complex maze from the image dots such as in Fig. 1. Fig.2 is the completed maze with the solution path drawn as red. Starting from 'S' circled, connecting all the image dots, and ending at 'E'. This path looks like a Hamiltonian path, which traverses all the image dots (vertices of a graph), each dot once. In addition, a Hamiltonian-like path is also constructed within every image dot. Hence the entire path could be fairly complex.

Third is to get a picture maze by applying the method to different pictures with some shaped items replacing some image dots. It enables the shapes to appear when the maze is solved.

4.1 IMAGE DOTS AS CONTOUR CYCLES AND PATH MAKING

Fig. 1 is the typical original picture from which a complex maze is constructed. The program with VC++ [16] for this study first recognizes each of the image dots of such pictures by extracting the contours of the dots, which form cycles and are numbered by OpenCV library [17] from 1 to n ($=30$) as shown in Fig.3. The cycles are organized with the maze path structure which is explained in section 1 and the previous works [9,11-13]. The connections between image dots are all based on these cycles. There are n dots in the picture. Then there should be n cycles with each having n_j length (number of vertices and edges for the cycle). A cycle, $cycle_j$, is a sequence of coordinates and

expressed as

$$cycle_j = \{ (x(j,i), y(j,i)) \mid 1 \leq i \leq n_j \} \quad (1)$$

The set of all cycles, CYCLES, extracted is expressed as:

$$CYCLES = \{ cycle_j \mid 1 \leq j \leq n \} \quad (2)$$

Each cycle is considered as a vertex of graph:

$$cycle_j = v_j \quad (3)$$

There may be a number of ways to make a closed path --- connecting all the dot cycles. One approach may be to connect one to a near one repeatedly until all are connected. This paper proposes a method to make a closed path connecting all the cycles once for each with no intersection between connecting lines. This mean that the path drawn is a plane graph. Details are described in the following subsections.

4.2 ALGORITHM

INPUT:

a set CYCLES of n image dot cycles by (2), Fig.3; each cycle is considered as a vertex (3).

OUTPUT:

a closed path P of length m (initially 0 and n at the end) with no intersection among the connecting lines (edges). P is comprised of a set of ordered cycles (4). But it represents an ordered set of edges. The last edge is by $cycle_n cycle_1$.

$$P = cycle_1 cycle_2 \dots cycle_n \quad (4)$$

Each cycle is in turn considered as a vertex. So P is an ordered set of vertices (5).

$$P = v_1 v_2 \dots v_n \quad (5)$$

DEFINITIONS:

The distance from $cycle_j$ to $cycle_k$ or v_j to v_k is defined as:

$$d(j, k) = \min_{1 \leq p \leq n_j \text{ \& } 1 \leq q \leq n_k} \sqrt{(x(j, p) - x(k, q))^2 + (y(j, p) - y(k, q))^2} \quad (6)$$

$P(j)$ is the j th item in P, which is circular.

METHOD:

1. Select an arbitrary $cycle_i$ from CYCLES; and delete it from CYCLES.
2. Find the nearest cycle from i and name it j and delete it from CYCLES
3. Find $cycle_k$ from CYCLES so that $d(j, k) + d(k, i)$ becomes minimal with no cross among connecting lines and cycles.
4. Organize P with constituent cycles i, k , and j which are v_i, v_k , and v_j and order them as 1, 2 and 3. Let the length of P be m ($=3$);
5. If CYCLES becomes empty, algorithm terminates with $m=n$ and draw P.
6. For all j from 1 to m and all q in CYCLES do {
 If ($j=m$) { $a=j$ and $b=1$ } else { $a=j$; $b=j+1$;}
 Find a, b and q so that $d(P(a), q) + d(q, P(b))$ becomes the minimal
 with no cross among connecting lines and cycles.
 }
 }
7. Insert q between P(a) and P(b), increment m and delete q from CYCLES. Go to 5.

CONDITION:

It is assumed that the sizes and the shape of contour cycles do not interfere the connections among them. This is usually justified.

EXPLANATION:

Since $d(P(a), q) + d(q, P(b))$ is the minimal, no cycle exists inside the triangle $(P(a), q, P(b))$, hence inside the region surrounded by P. This is explained later. Even if $d(P(a), q) + d(q, P(b))$ is the minimal, there is a possibility that the connecting lines intersects. So selection of q is always made under the condition that no intersection occurs. It is naturally justified that at least one contour cycle always exists to be inserted to P if CYCLES is not empty because P is the closed path, circular hence faces in all directions.

Fig. 4 shows that an arbitrary $cycle_{16}$ is selected and $cycle_{22}$ is found as the nearest and they are connected. Fig. 5 shows that $cycle_{19}$ is selected because $d(16, 19) + d(19, 22)$ is the minimal and form $P(cycle_{16}, cycle_{19}, cycle_{22})$ as clockwise. Fig. 6 illustrates that $cycle_{26}$ is selected and inserted in P and the path becomes $P(cycle_{16}, cycle_{19}, cycle_{26}, cycle_{22})$. Fig. 7 is the final closed path which visits all cycles, each only once without intersection.

ALGORITHM PROOF:

The algorithm proof is given by induction:

When there are three ($i=3$) cycles in P, P is C_3 , a triangle with no cycle inside. Suppose that P is C_r and no cycle inside the closed path when $i = r$. As long as CYCLES is not empty, there

248 is a cycle in CYCLES, named as q such that $d(P(a), q) + d(q, P(b))$ becomes the minimal with
 249 no intersection. This is guaranteed because P is circular so it faces in all directions. The cycle
 250 q is the $(i-r+1)$ th cycle deleted from CYCLES and it is the $(r+1)$ th cycle inserted in P . Therefore,
 251 P becomes C_{r+1} . This completes the proof
 252
 253
 254
 255

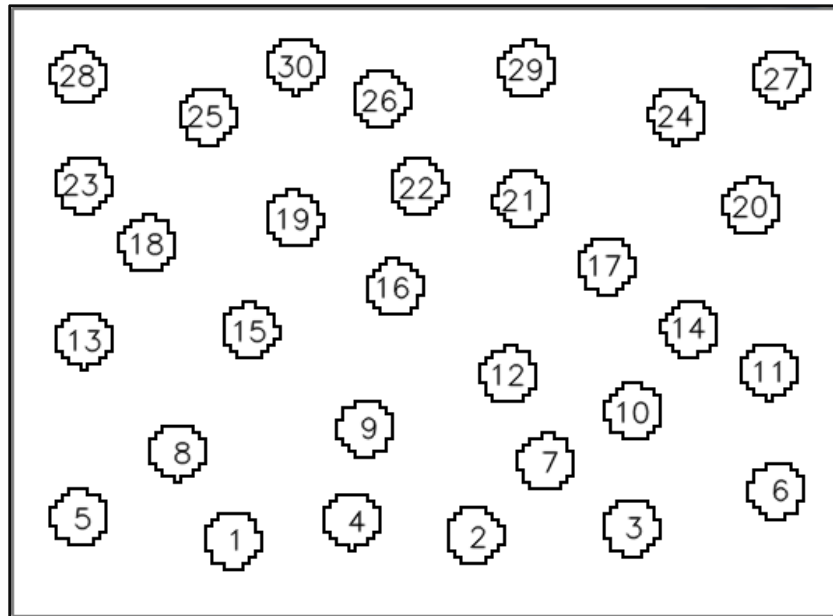


Fig.3 Dot contour cycles

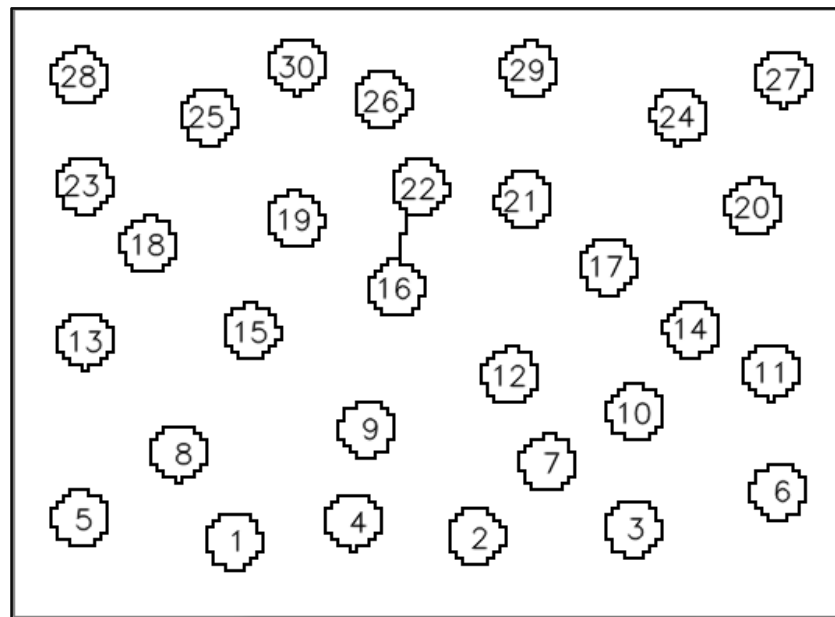


Fig.4 First connection

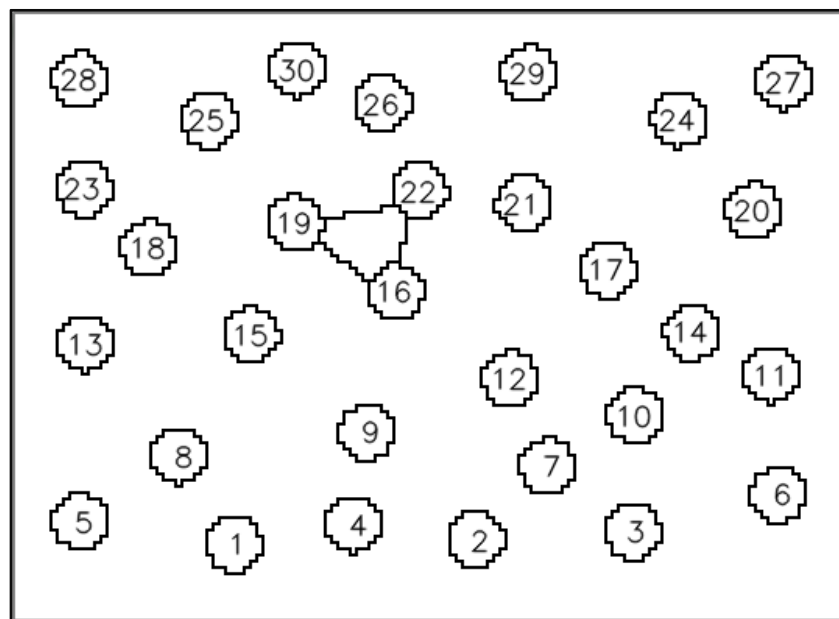


Fig. 5 Three dot cycle closed path

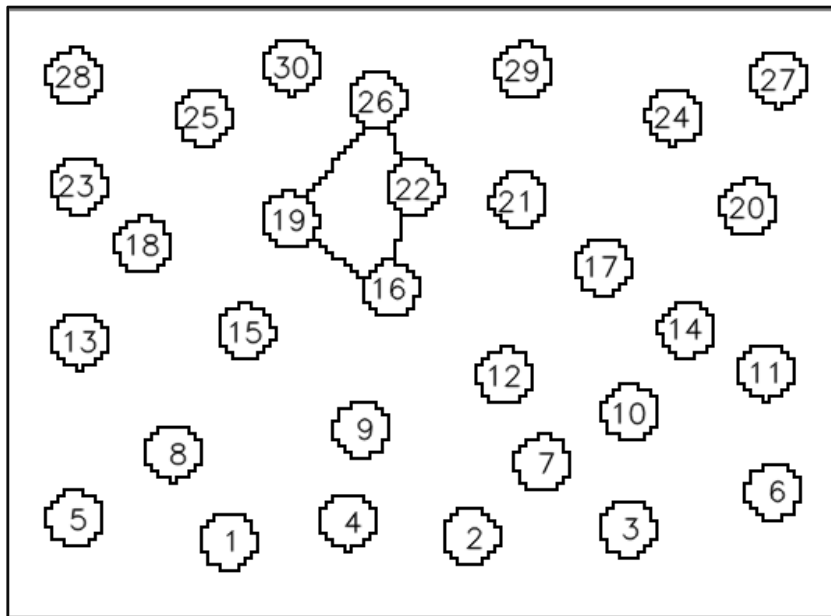


Fig.6 No. 26 is inserted in the closed path

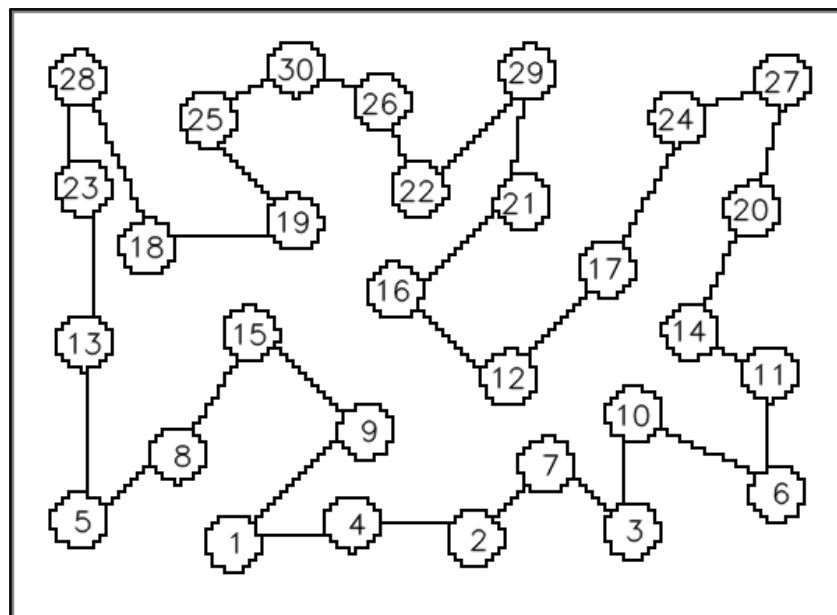


Fig. 7 Closed path completed

5 OBTAINING COMPLETE MAZE AND DISCUSSIONS

5.1 MAZE CONSTRUCTION ON A NUMBER OF IMAGE DOTS

After constructing the closed path (Fig.7), the following steps were applied. The line between two cycles (16, 21) was removed to designate the start and the goal. The path from the start cycle (16) to the end (21) became one stroke line. Whenever a cycle was encountered going along the path, an arc-like curve was randomly chosen. The result is shown as in Fig. 8. Within an image dot region, Hamiltonian-like path was organized. This was done by successive insertions of path parts using the method [9, 10, 14]. Almost all points (vertices) within an image dot were inserted into the path, but not necessarily all. At this point, the solution path of the maze was competed (Fig. 9). Then, random dead end branches were added and walls of the maze were thinned to finish the maze (Fig. 10). Fig. 2 is the maze with the solution path drawn red as mentioned before. Fig. 11 is another completed maze with the solution. The connection lines in Fig. 11 were randomly curved a little. This is to hide the solution more effectively.

5.2 CONSTRUCTION OF PICTURE MAZE

The same procedure was applied to the image items of Fig. 12. The obtained maze is a complex picture maze. Fig.13 shows the maze with the solution. This maze is a quiz to find an English word "COMPUTER". Accordingly, when the player draws the solution path, letters appear---'C', 'R', 'O', 'E', 'U', 'P', 'T', and 'M'. The player is naturally asked to find an English word, "COMPUTER" using the found letters. The connecting lines between image items are with some randomized zigzag so that the lines on the maze cannot be seen easily before it is solved.

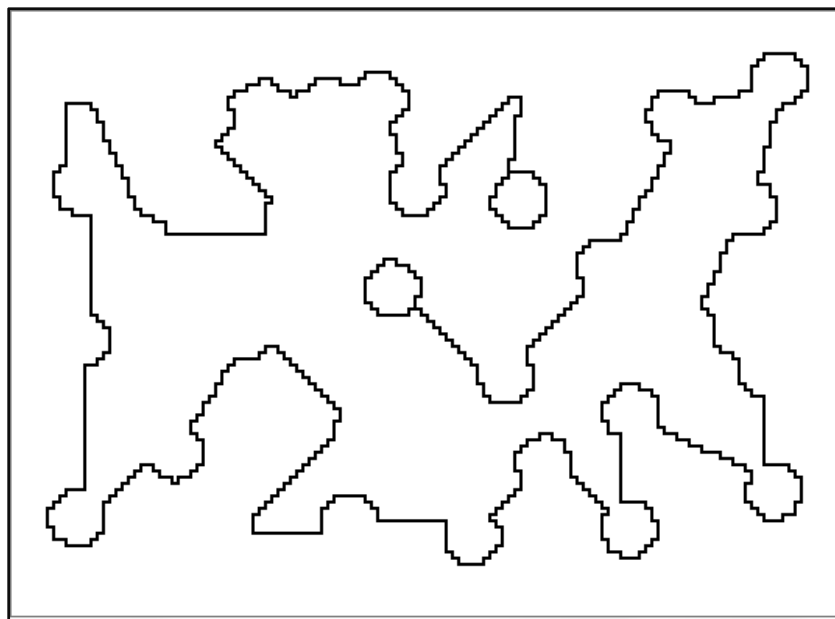


Fig.8 Starting base line of solution path

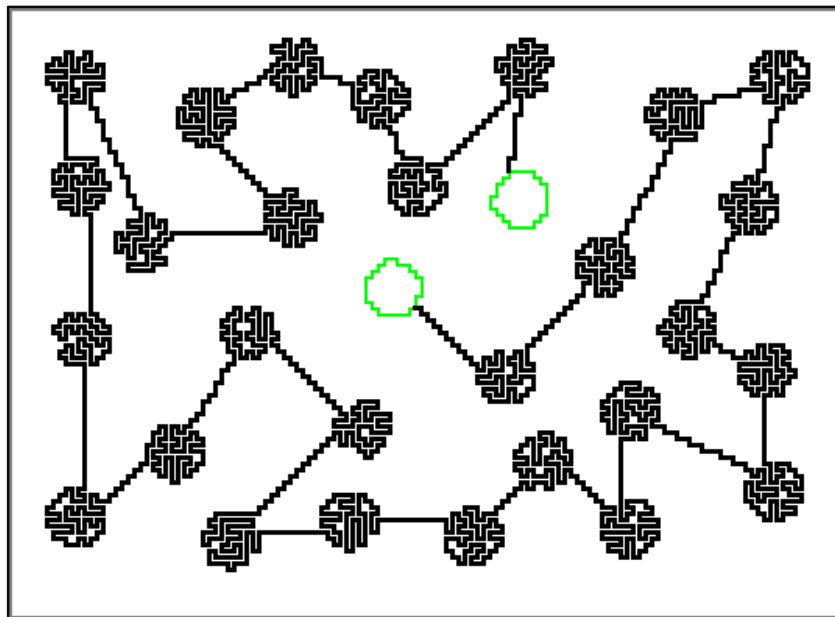


Fig.9 Maze solution path completed

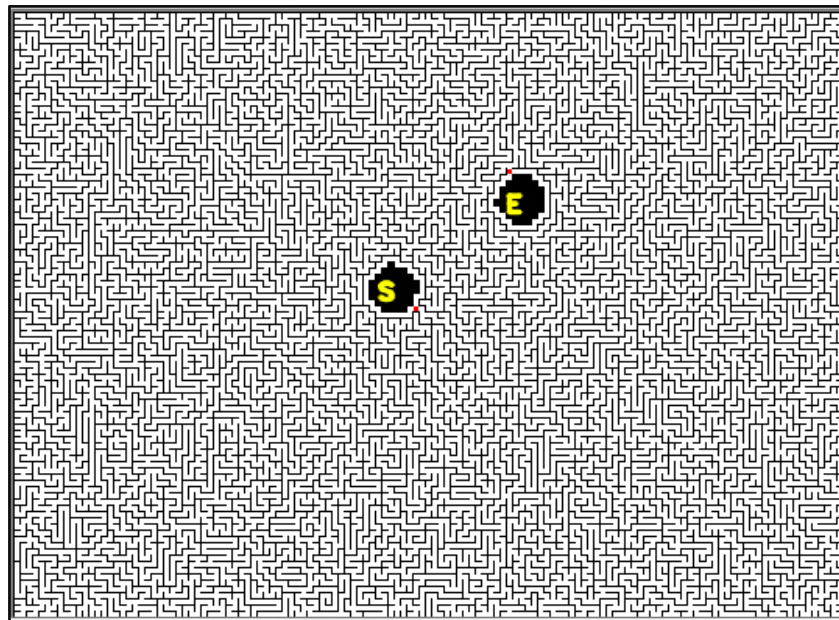


Fig.10 Final maze completed to be presented to players

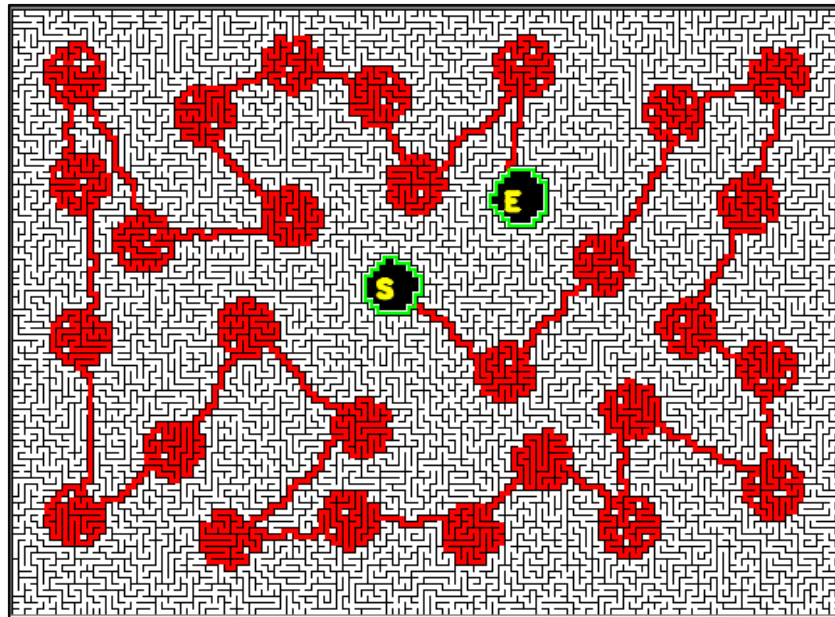


Fig.11 Final maze but with curved solution path lines

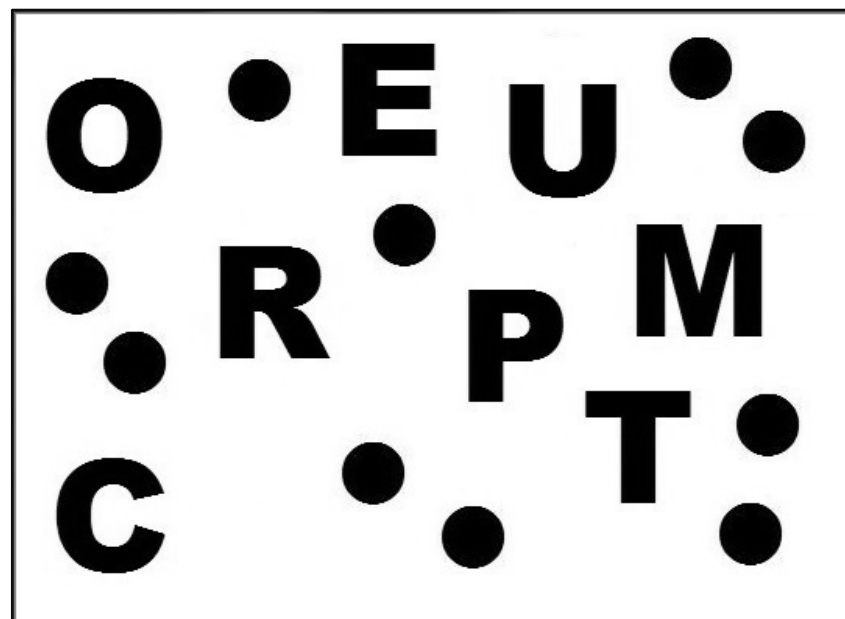


Fig. 12 Original binary picture with the alphabets of "COMPUTER"

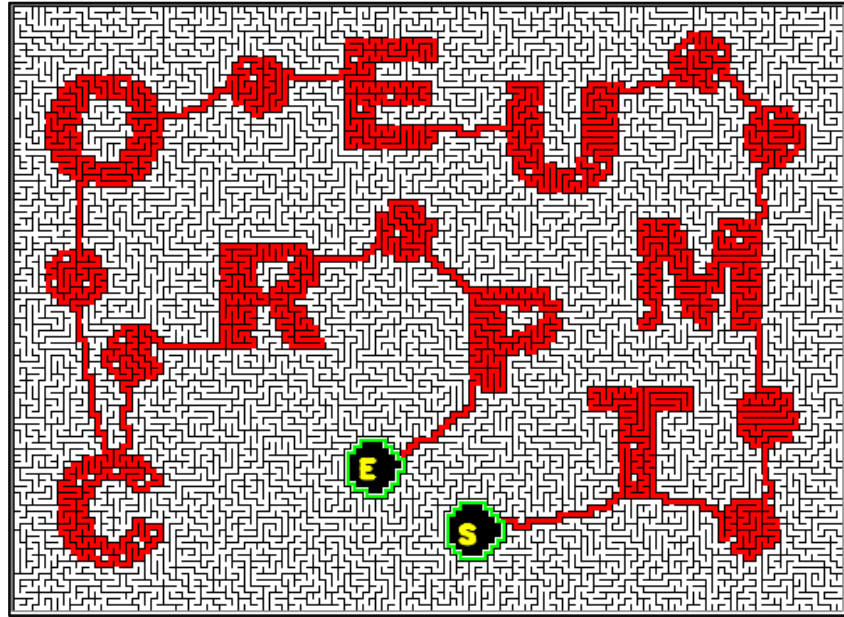


Fig. 13 Complete quiz maze with solution path to find "COMPUTER"

5.3 ABOUT THE PROCESS OF ORGANIZING P

P is the closed path of a sequence of the vertices which traverses each of the vertices extracted from a given picture once for every vertex. Whenever one vertex is inserted in P by the algorithm previously stated, there appears a situation illustrated in Fig.14. Consider the triangle organized by the three vertices $P(a)$, q , and $P(b)$ as in Fig.14, where q is a vertex in CYCLES and b is the next item number of a in P and the triangle gives the minimal sum of two edge lengths $d(P(a), q)$ and $d(q, P(b))$. If there is a vertex q' inside the triangle from CYCLES, then the following inequality, which is well known, holds.

$$d(P(a), q) + d(q, P(b)) \geq d(P(a), q') + d(q', P(b)) \quad (7)$$

This means that there is no vertex inside the triangle if $d(P(a), q) + d(q, P(b))$ is the minimal for a, b in P and q in CYCLES. If the combination of a, b and q always provides the minimal, then there is no vertex inside the region surrounded by P. Therefore, we can continue to look for such vertices outside the region until CYCLES becomes empty. By the way, we have the following inequality:

$$d(P(a), q) + d(q, P(b)) = d(P(a), q) + d(q, r) + d(r, P(b)) \quad (8)$$

$$\geq d(P(a), r) + d(r, P(b)) \quad (9)$$

$$\geq d(P(a), q') + d(q', r) + d(r, P(b)) \quad (10)$$

$$\geq d(P(a), q') + d(q', P(b)) \quad (11)$$

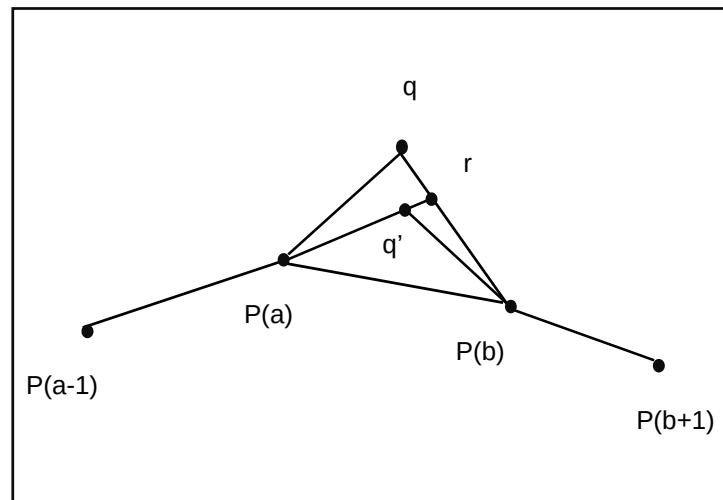


Fig. 14 Triangle ($P(a)$, q , $P(b)$). b is the next item number of a in P which is circular.

6. CONCLUSION

The following are the conclusions:

1. Analogy is naturally made between image processing and graph theory or between image space and graph space.
2. The proposed algorithm showed that it can construct a path traversing all the image dots once for each. The path looks like a Hamiltonian path. The algorithm is proved by induction. The essential point, which is different from ordinary graph drawing, is that the positions of image dots (vertices) are predetermined. Graph is drawn on these prepositioned dots (vertices).
3. It is shown that a complex maze can be constructed using the method. If shaped images are used, the maze becomes a picture maze.

REFERENCES

- [1] T. Hosokawa, "Maze," <http://aanda.system.to/> Accessed 2013/01/25
- [2] "Maze generation algorithm," http://en.wikipedia.org/wiki/Maze_generation_algorithm, Accessed 2013/01/09
- [3] J. Xu and C. S. Kaplan, "Image-Guided Maze Construction," ACM Transactions on Graphics, 26, 3, Article 29, pp.1-9, 2007
- [4] J. Xu and C. S. Kaplan, "Vortex maze construction," Journal of Mathematics and the Arts, 1, (1), pp.7-20, 2007
- [5] Conceptis puzzles, "Maze-a-PiX," <http://www.conceptispuzzles.com/>, accessed 2013/01/09
- [6] Y. Okamoto and R. Uehara, "How to make a picturesque maze," Proc. Canadian Conf. on Computational Geometry, pp.137-140, 2009

- 648 [7] H. Hamada, <http://www.lab2.kuis.kyoto-u.ac.jp/~itohiro/Games/100301/100301-11.pdf> ,
649 accessed 20013/01/12
- 650
- 651 [8] K. Hamada, "A Picturesque Maze Generation Algorithm with Any Given Endpoints," Journal
652 of Information Processing, Vol. 21, No. 3, pp.393-397, 2013
- 653
- 654 [9] T. Kurokawa, "Picture Maze Generation by Successive Insertion of Path Segment," British
655 Journal of Mathematics and Computer Science, Vol.4, Issue.24, pp.3444-3463, 2014.12,
656 Science Domain international, ISSN:2231-0851,
657 <http://www.sciencedomain.org/issue.php?iid=699&id=6>
658
- 659 [10] K. Ikeda and J. Hashimoto, "Stochastic Optimization for Picture Maze Generation," Jouhou
660 Shori Gakkai Ronbunshi, Vol. 53, No.6, pp.1625-2634, 2012, in Japanese
- 661
- 662 [11] T. Kurokawa, K. Mori, and T. Mizuno," Automatic construction of picture maze by repeated
663 contour connection," Thailand-Japan Joint Conference on Computational Geometry and
664 Graphs, pp.5-6, 2012
- 665
- 666 [12] T. Kurokawa, "Picture maze generation and its relation to graphs," 2013 SHIKOKU-SECTION
667 JOINT CONVENTION RECORD OF THE INSTITUTE OF ELECTRICAL AND RELATED
668 ENGINEERS, p.231, 2013
- 669
- 670 [13] T. Kurokawa, "Picture Maze Generation by Repeated Contour Connection and Graph Structure
671 of Maze," Computer Science and Engineering, Vol.3, No.3, pp.76-83, 2013.12, Scientific &
672 Academic Publishing, USA , ISSN:2163-1484
673 <http://article.sapub.org/10.5923.j.computer.20130303.04.html> Accessed 2014/1/20
674
- 675 [14] F. J. Wong and S. Takahashi, "Flow-based Automatic Generation of Hybrid Picture Mazes,"
676 Computer Graphics Forum, 28, (7), pp.1975-1984, Wiley-Blackwell, 2009
- 677
- 678 [15] T. Kurokawa, "Maze Construction by Using Characteristics of Eulerian Graphs," British
679 Journal of Mathematics & Computer Science, Vol.9, Issue.6, pp.453-472, 2015.6, SCIENCE
680 DOMAIN international, UK, ISSN: 2231-0851, <http://sciencedomain.org/issue/1147>
681
- 682 [16] <http://www.microsoft.com/ja-jp/dev/express/default.aspx>
683
- 684 [17] <http://jaist.dl.sourceforge.net/project/opencvlibrary/opencv-win/2.3/> Accessed Oct., 2011