Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Standard Jigsaw Puzzle Solver

[1]Monika Jingar*, [2]Bachcha Lal Pal
[1]M-Tech (CS) Scholar, [2]Assistant Professor and HOD
Department Of Computer Science, Mewar University,
Rajasthan, India

Abstract—In this paper we introduce an algorithm build for solving standard jigsaw puzzle problem which is the most popular and mostly used variant of jigsaw puzzle. The problem is known to be an NP-complete problem. The algorithm can differentiate standard jigsaw puzzle tiles from its other variants such as shape along, rectilinear, square puzzle etc. It uses two step verification process 1st is to check if the given tiles are of standard jigsaw puzzle of not and 2nd is to actually interlock those tiles together in order to solve the problem.

Keywords— Standard jigsaw puzzle, algorithm, image reconstruction, puzzle solving, data set.

I. INTRODUCTION

Jigsaw puzzle was 1st introduced in around 1760s by European cartographer John Spilbury who named it "dissected puzzle". It got its famous name "jigsaw puzzle" because they were built by painting pictures on large sheets of wood, then cut into pieces using a jigsaw [1][2]. Ever since then a lots of variation of this type of tile puzzle are introduced with one thing in common which is interlocking the given tiles in order to solve it. The most famous one is standard jigsaw puzzle which everybody once in their life time has solved [3]. The element which make it different from the other variants is that in this type of puzzle each indent perfectly interlock with each outdent, the only way to distinguish the right tile with others is the combination of those indent, outdent and null side, its pictorial information and their sequence match with each corresponding tiles. Typically a standard jigsaw puzzle got 4 sides as its basic shape can be a square or a rectangle depending on the manufacturer.

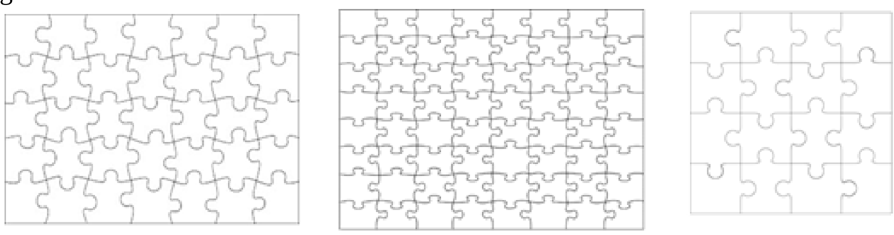


Fig. 1: Standard jigsaw puzzle templates

A number of papers have been written around the topic of jigsaw puzzle and their solutions, but the majority was of edge-matching jigsaw puzzles and shape-along jigsaw puzzles. But very few for pictorial jigsaw puzzle. This problem is considered as one of the NP-complete problem [4][5] and as every problem lies in this category can be solved using same approach such as reconstructing of archaeologicalartifacts [8][9], shredded documents, fitting together surface patches to a scanned objects, genomes sequencing [7], cutting stock problem and many more [6]. And this is how all variants of jigsaw puzzles are connected together if we remove indent ad outdent from the tiles then the puzzle will become edge-matching jigsaw puzzle or removing the pictorial information and changing the size and orientation of indent and outdent will make shape-along jigsaw puzzle in other words any problem which deals with a number of pieces and reconstructing something meaningful is jigsaw puzzle.

Fig. 2: Shape-along jigsaw puzzle and edge-matching jigsaw puzzle

The standard jigsaw puzzle problem has two main difficulties which is geometry basically all indent and outdent are located in the middle of each side and in most cases they are identical i.e. each indent will interlock with each outdent unlike shape-along jigsaw puzzle where there is only one possible indent for each outdent and vice versa, second is combinatorial which is that there are a lots of number of combinations available for any tile, to tackle this we can use all the information given in a piece of tile which is its shape, number of indent, outdent and null side and the pictorial information will help us in determining if a corresponding tile is the perfect neighbour , but we have to rotate a tile because there are four sides and also the orientation was unknown for a given tile [10].

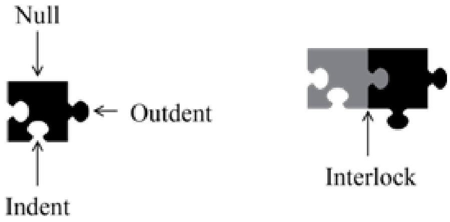


Fig. 3: Indent, outdent, null and interlock of two tiles

II. Overview

There are certain standard rules which any puzzle solver follows while solving a jigsaw puzzle, which are made on basis of general understanding while dealing with this kind of problems. In order to solved a standard jigsaw puzzle a solver will 1st separate tiles with null side, then try to form a frame with the help of those null side tiles and then start joining the rest of the tile by using the picture provided by the manufacturer which is most of the time used to be the packet/box in which the tile are packed.But this problem become harder when there is no reference picture provided, in this case a solver uses the color as the guidance and try to fit the right piece in right place with try and error method and continue it till all the tiles are joined together.

The algorithm presented in this paper is divided into four parts, part one include autorotation of tiles, part two finding total numbers of indent, outdent and null sides, part three condition check for all tiles and part four mainly deals with merging and matching of each tiles in order to solve the puzzle.

III. Dominating Slope and AutoRotation

When we scan the tiles it is not necessary that they're in well align orientation, but to make the upcoming steps huddle free we introduce dominating slope and autorotation which will be performed in each tile to align them in right orientation. For this we 1st scan each tile from the top then all pixel values as seen from the top are noted in an array. A slope will contain simultaneous increasing /decreasing values. Local sets of slopes are computed and the largest in pixel number are taken for autorotation computation. Once a dominating slope is determined, a first degree polynomial trendline is constructed taking the pixel positions into consideration giving an equation such as $y = m x + c$. Our goal for the puzzle tile to be straight must contain the equation $x = c$ or $y = c$. The angle between the 2 equations are determined using the tan---1 formula and the puzzle block is rotated.

Fig. 4: A 9 tile standard jigsaw puzzle

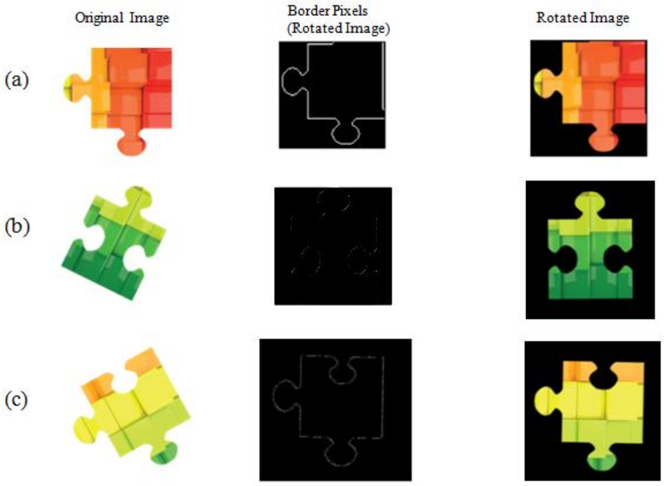


Fig. 5: Tiles before and after rotation

The tile used in (a) is already in correct alignment this is why after rotation it looks the same, whereas tile in (b)'s dominant slop is towards right and the algorithm detected it and rotate it towards left to align it well, same procedure is applied on the tile in (c)except its slope is towards left and was rotated towards right to make the slope a straight line and to balance the equation.

IV. FINDING BORDER AND DENTS

After rotation, finding the given puzzle tile's border is our next step, its pixel position and values (R,G and B channels) are crucial for indent, outdent or null determination and matching. Again, pixel positions are computed as seen from the top in an array. A relaxation of 3 is granted. That means maximum and minimum values in the array are computed from array position 2 to n---3and the distance is the difference in pixel positions of the same.

If maximum > dominating slope pixels + 3 := Outdent
Else if minimum< dominating slope pixels – 3 := Indent
Else Null

Once, indent, outdent or null has been determined, they are stored in a 2D array similar to the image below.

				Indent/ Outdent/ Null
				Distance "d"
				Distance "x" Position
				Distance "y" Position

Fig. 6: 2D array and its content

V. CONDITION CHECK

In this step we study each tiles and check if they fit in the condition of being a tiles of a standard jigsaw puzzle. Due to the fact that each standard jigsaw puzzle follows a well-defined sets of tiles. We classify the tiles and puzzle's property and make the conditions for each tiles, which are as following:-

Tile and its property:
T = {i,o,n}
Where
T = tile, i = indent, o = outdent and n=null
s =min size of the puzzle (required no. of tiles to form a puzzle)
x=degree of puzzle (max connection possible for one tile)

Condition 1:
A tile which has a minimal number of 2 indent+outdent should be valid tile.

$$i+o>=2 \qquad \ldots 1$$

Condition 2:
Total number of i,o and n should be equal to 4

$$i+o+n=4 \qquad \ldots 2$$

Condition 3:
Min size of the puzzle should be 4 (derived from condition 1)

$$s>=4 \qquad \ldots 3$$

Condition 4:
Degree of the puzzle will be equal to 4

$$x==4 \qquad \ldots 4$$

Following all the above conditions possible tiles of one standard jigsaw puzzle can be as Fig:8 but the orientation of each tiles can be different in terms of their location of indent, outdent and null sides on a tile.

VI. MATCHING AND MERGING

Matching and merging will start by choosing any two random tiles and it will archive by following steps:-

i. Match for d
ii. Match borders
iii. Match colors

As illustrated below "d" is the mid of indent/outdent also the highest depth/height of them, we knew that once interlock will only occur with one indent and one outdent in two different tiles. Once the "d" is detected, we then match both tile's d together, which will result in a perfect interlock.

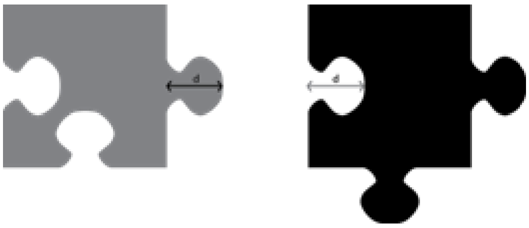


Fig. 7: "d" in indent and outdent

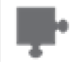
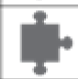
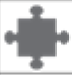
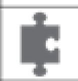
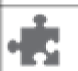
S.no.	Puzzle Tiles	Indent	Outdent	Null
1.		0	2	2
2.		0	3	1
3.		0	4	0
4.		1	1	2
5.		1	2	1
6.		2	0	2
7.		2	1	1
8.		2	2	0
9.		3	0	1
10.		3	1	0
11.		4	0	0
12.		1	3	0

Fig. 8: Tiles which fits the conditions

VII. Border Color Match

After shapes match, border colors are matched. In this step, as shown in fig. 5, border color pixels of each individual red, green and blue in RGB pixel information is taken and the difference of each edge to another block's edge to determine a good fit. Each individual block being 200 by 200 pixels each. Fig. 9below, shows the red channel information of the same shapes in fig. 4 of part of the border taken – one indent and 2 outdents which have shapes match.

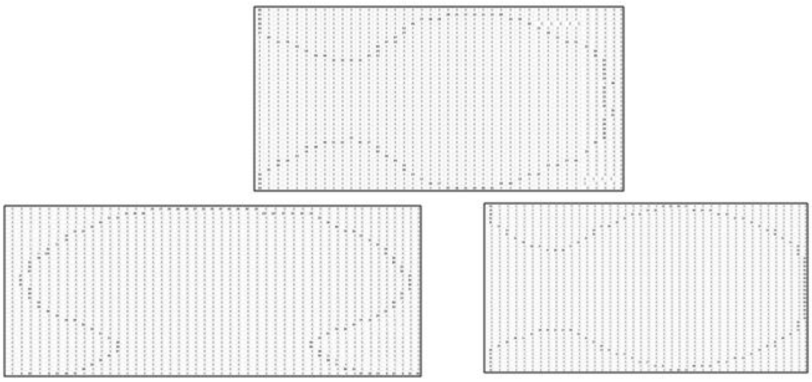


Fig 9. Number matrix of one tile's indent and outdents

Similarly for the edges.

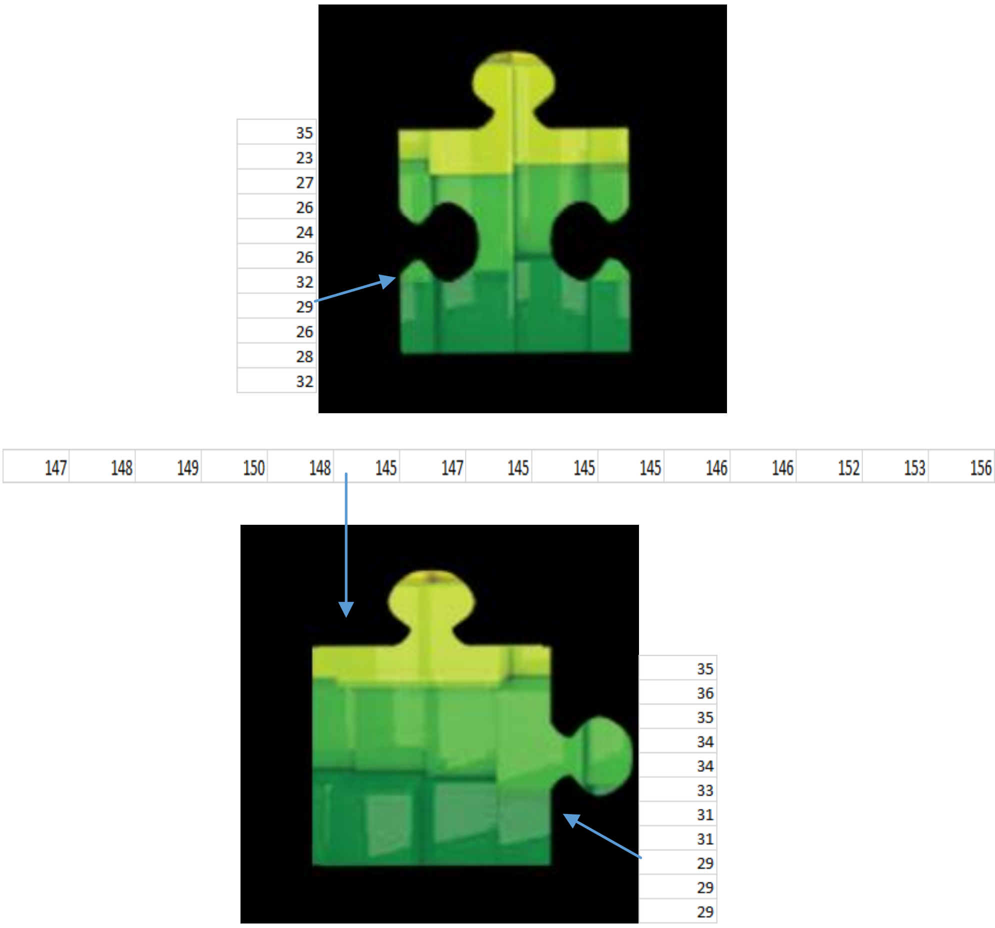


Fig 10. Border pixel's matrix of tiles

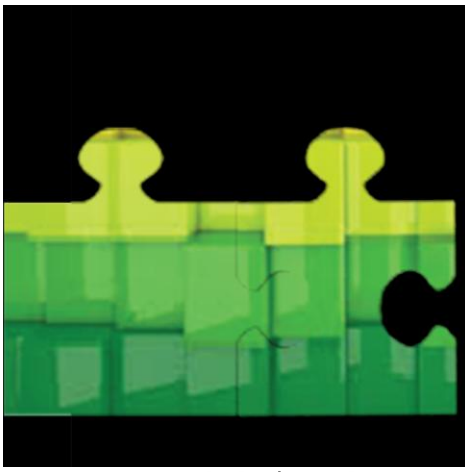


Fig 11. Interlock of two tiles

Similar process is done for green and blue channels before merging. Execution time taken was 23.76seconds yielding us the result.

VIII. CONCLUSIONS

The algorithm proposed in this paper is designed to deals with one of the most famous np-complete problem. The steps taken before even starting to solve the actual problem helps to reduce the times of error occurring which one can face during the actual solution process. As all problem lies in the category of np-complete problem are somehow related to each other, proposedalgorithm can workfor edge-matching and shape-along jigsaw puzzle after few logical altering in the condition which are suitable while dealing with such problem. The condition for each tiles can also help to eliminate those unexpected tiles which sometimes come with a pack of jigsaw puzzle (mostly a scrap piece of board) during packaging process. This algorithm is tested in puzzle tiles varies from 4-32 showing possibility of solvinglarge number of standard jigsaw puzzle.

REFERENCES

[1] Andrew C. Gallagher, "Jigsaw Puzzles with Pieces of Unknown Orientation",12 Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Pages 382-389.

[2] H. Freeman and L. Gardner. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. IEEE. Trans. on Electronic Computers, 1964.

[3] Anne D. Williams. Jigsaw Puzzles: An Illustrated History and Price Guide. WallaceHomestead.

[4] T. Altman. Solving the jigsaw puzzle problem in linear time. Applied Artificial Intelligence an International Journal, 3(4):453–462, 1989.

[5] E. Demaine and M. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. Graphs and Combinatorics, 23:195–208, 2007.

[6] Anne D. Williams. The Jigsaw Puzzle: Piecing Together a History. Berkley Books, New York, 2004.

[7] W. Marande and G. Burger. Mitochondrial DNA as a genomic jigsaw puzzle. Science, 318(5849):415–415, 2007.

[8] B. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, C. Doumas, S. Rusinkiewicz, and T. Weyrich. A system for high-volume acquisition and matching of fresco fragments: Reassembling theran wall paintings. ACM Transactions on Graphics, 27(3):84, 2008.

[9] D. Koller and M. Levoy. Computer-aided reconstruction and new matches in the forma urbisromae. Bullettino Della CommissioneArcheologicaComunale di Roma, pages 103–125, 2006.

[10] D. Goldberg, C. Malon, and M. Bern. A global approach to automatic solution of jigsaw puzzles. Computational Geometry: Theory and Applications, 28(2-3):165–174, 2004.