A Method for Automatic Detection of the Square Piece In The Borderless Square Jigsaw Puzzle

Sendi Novianto *sendi.novianto@dsn.dinus.ac.id*
School of Automation
South China University of Technology
Guangzhou - Guangdong, 510640, China

Luo Fei *aufeiluo@scut.edu.cn*
School of Automation
South China University of Technology
Guangzhou - Guangdong, 510640, China

Abstract

This research is a continuation from previous research, which is focused on labeling and finding missing pieces of jigsaw puzzle which has square-shaped of puzzle pieces, and each piece has a border surrounding it. The Aim of this research focuses on the finding missing pieces from the jigsaw puzzle, on the other hand the jigsaw puzzle image that we use does not have surrounded-border in each piece. For small sizes of the puzzle as well as 3 x 3 (9 pieces), the process for searching the missing pieces with its position can be done manually, conversely when the size of the puzzle is more than 100 pieces, the searching process manually will take some times. Our contribution is detecting automatically for finding pieces with position, size of each piece and total pieces in the jigsaw puzzle that has 25%, 50%, 75% and 99% of missing pieces in the image of borderless square jigsaw puzzle. The methods we use are ”Blob Analysis” and combine with a line search based on a column used to separate square pieces which have a different size, we call this method as BALSEM (Blob Analysis with Line SEarch based on a coluMn). The results of this research demonstrate the success of the process of combined-methods we use.

Keywords: Square Piece, Jigsaw Puzzle, Borderless Piece.

1. INTRODUCTION

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, cars and other things) in digital images and videos. In object detection algorithms typically use extracted features and learning algorithms to recognize instances of an object category. Well-researched domains of object detection include face detection and pedestrian detection. Object detection especially in image analysis has applications in many areas of computer vision, including puzzle detection [1], counting of bacterial [2], enumeration of malaria parasites [3], diabetic retinopathy [4], classification of white blood cells [5], facial wrinkles [6], noisy logo recognition [7], Water Quality Analysis [8], Eye detection in a face image [9], Fingerprint classification [10].

Every object in the image have some bounding pixel one to another. To find the minimum-bounding also affords various forms of normalization and to match some object as well. This plays important roles in applications involving industrial parts inspection and assembly, object detection from remote sensing imagery and biomedical image analysis, blob analysis [11], [12]. Connected-component labeling consists in traversing an image and determining regions whose pixels share similar properties, such as intensity values, and are connected with each other. Furthermore, all displace regions are identified and each connected-component is signed to a unique label. The extraction and labeling of connected components is a crucial step to several image analysis applications, such as character recognition, fingerprint identification, medical image segmentation [13], robot vision, among others [14].

Jigsaw puzzles are one of the games that are on the computer. The rules to play this game is to put the pieces by pieces together becomes a picture called intact. This game will not be done if all the pieces are not yet installed. Sometimes, when we play this game, there are still one or more pieces are missing and cause the game cannot be completed. This research focuses on the detection of missing pieces, so the computer can detect the number of the missing pieces accompanied by the position of the missing pieces in the game.

Hongsheng Li et al. try to solve aspecial type of jigsaw puzzles, they try to reconstructing banknotes from a large number of fragments based on frafments' images. A main limitation of the methods is that they do not leverage the following important observations: an intact banknote's image is known and thus can be used as prior information, and if two aligned fragments overlap each other, they must not be from a same banknote. based on these two important ovservations, a three-step method is proposed to reconstruct banknotes from their fragments. each fragment is first aligned to its original position on the banknote by RANSAC method. after evaluating every two aligned fragments' relationship, all fragments are embedded into a lower dimensional space and then clustered into small groups using a modified agglomerative clustering method. fragments in a same cluster are likely to be from a same banknote. experiments on both synthetic and real data demonstrate the efectiveness of their proposed method [15].

Michael makridis et al, on his research, they propose a new technique for solving jigsaw puzzle. the novelty of the propsed technique is that it provides an automatic jigsaw puzzle solution without any initial restriction about the shape of pieces, the number of neigbord pieces, etc. the proposed technique uses both curve and color matching similarity features. a recurrent procedure is applied, which compares and merge puzzle pieces in pairs, until the original puzzle image is reformed. geometrical and color features are extracted on the characteristic point of the puzzle pieces (CPs) of the puzzle pieces. CPs, which can be considered as high curvature point, are detected by a rotationally invariant corner detection algorithm. the features which are associated with color are provided by applying a color reduction technique using the kohonen selft-organized feature map. finally a postprocessing stage checks and corrects the relative position between puzzle pieces to improve the quality of the resulting image. experimental results prove thee efficiency of the proposed technique, which can be further extended to deal with even more complex jigsaw puzzle problems[16].

Jinn-Tsong Tsai et al, study implement a genetic algorithm (GA) with the condensed encoding and the improved fitness function is used to solve japanese puzzle. in this study, the condensed encoding can make sure that the chromosome is a feasible solution in rows for japanese puzzle. in the reconstruction process of a Japanese nonogram, the number in the left column are used as encoding conditions, and the cnumbers in the top row with the improved fitness function are employed to evaluare the reconstruction result[17].

Jigsaw puzzle solving is necessary in many applications. including biology, archaeology, and every day life. Genady Paikin Technion et al try to reconstruct the image from a set of non-overlapping, unordered, square puzzle part. their contribution is a fast, fully-automatic, and general solver, which assume no prior knowledge about the original image. it is general in the sense that it can hendle puzzle of unknown size, with pieces of un known orientaion,and even puzzle with missing pieces. moreover, it can handle all the above, given pieces from multiple puzzle. through and extensive evaluation we show that our approach outperforms state-of-the-art methods on commonly-used datasets[18].

In the previous research, the researchers generated the missing pieces and used comparation technique to the missing pieces position in order not to produce the other missing pieces in the same area. Besides, in the process of missing pieces detection,the researchers used the border lines in every puzzle's pieces to facilitate the detection of the missing pieces[1]. In this research, the process of missing pieces generation selects and takes the order of puzzle's pieces in shuffled mode. In addition, in the process of finding the missing pieces, the researchers use the BALSEM method.

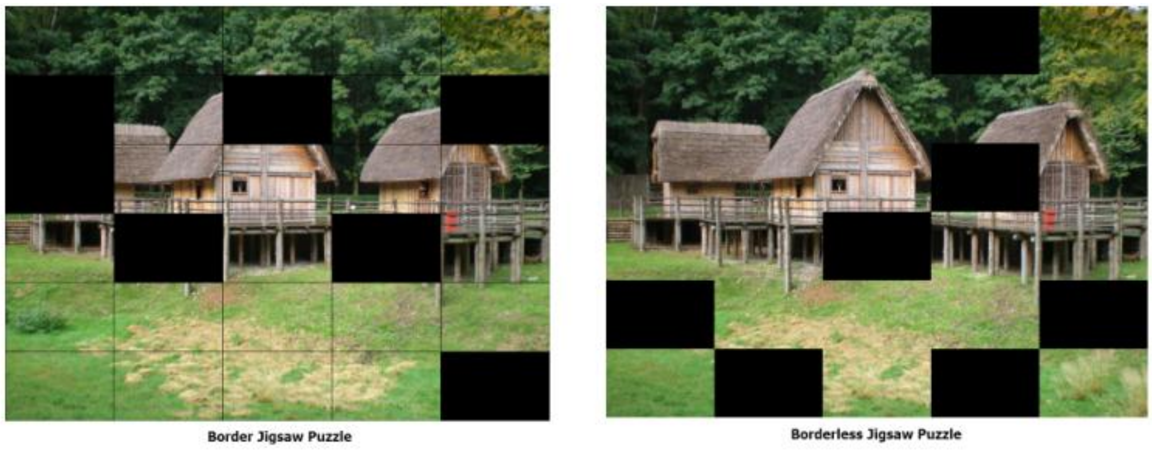

FIGURE 1: Jigsaw Puzzle pieces with border and pieces without border.

This research is divided into several stages, as follows:

- Borderless Jigsaw Puzzle Image Formation.
  a. Requirement input parameter that includes image input in RGB format (I_I), how many columns (C_P), how many rows (R_P), and how many missing pieces (M_P) that we want to be generate.
  b. Detection of width (W_I) and height (H_I) of (I_I).
  c. New initialization in width (W_I) and height (H_I) of (I_I).
  d. Detection of width (W_P) and height (H_P) of square piece (SQ_N) including width and height of missing piece (M_P).
  e. Replace all squares include missing piece squares (C_{PM}, R_{PM}) on the (I_I) to become full image of jigsaw puzzle in RGB type (I_{JP}) with its information of width (W_{JP}) and height (H_{JP}).

- Converting and detecting color.
  a. Converting color in the image input (I_{JP}) that have RGB value more than 0 (black color) to RGB value 255 (white color) (I_{JPW}).
  b. Converting RGB image type (I_{JPW}) become grayscale image (I_{JPG}).
  c. Get all the color in (I_{JPG}) and reversed it (I_{JPR}).
  d. Get all of the black color in (I_{JPG}) with location [B_{IJPG} (x, y)] and (I_{JPR}) with location [B_{IJPR} (x, y)].

- Assembling the square.
  a. Assemble the square, it is a process made from the information on [B_{IJPG} (x, y)] and [B_{IJPR} (x, y)], we start to assemble vertical line from the same y absis position (L_N). When it finds the space between one position with next position in y absis position, it assumed as new vertical line (L_{N+1}).
  b. From vertical line (L_N), the next step is combining all vertical line with the same height become one square (SQ_N), if there is different height, it is assumed as new square (SQ_{N+1}).
  c. After getting all squares, we will try to find the smallest width (W_{SQ}) and height (H_{SQ}) for each square (SQ_N).

- Merging and Labeling.
  a. Dividing (W_{JP}) with (W_{SQ}) to get how many square in column (C_{JP}) and (H_{JP}) with (H_{SQ}) to get how many square in row (R_{JP}).
  b. Arranging all the squares in each (C_{JP}) and (R_{JP}) and detecting how many color inside the square, if the color inside the square more than 0 it means that the color has piece inside the square (P), if just 0 it means the missing piece is inside the square (M).
  c. Give the label in each square (SQ_N) in (I_{JP}) include position of (Column, Row) and count it, how many pieces are not missing and missing pieces inside (I_{JP}) as well.

2. BORDERLESS JIGSAW PUZZLE IMAGE FORMATION

The image sources used in this research is from [1]. Parameter input for this process is Image Input (I_I), how many row (R_P), how many column (C_P), and how many missing pieces that will be created (M_P). From (I_I) we can get property of width (W_I) and height (H_I) of (I_I). Before start to create both column and row in this jigsaw puzzle, we will begin to make sure the (W_I), if mod by (C_P) the result is 0 or not "Eq. (1)", it is applied to (H_I) with (R_P) "Eq. (2)" as well. To get the width (W_P) of pieces, we divide (W_I) with (C_P) "Eq. (3)", and to get height (H_P) of pieces, we divide (H_I) with (R_P) "Eq. (4)".

$$if\,(W_{\mathrm{I}}\,mod\,C_P)\begin{cases} = 0, & W_I = W_I \\ > 0, & W_I = W_I - (W_{\mathrm{I}}\,mod\,C_P) \end{cases} \tag{1}$$

$$if\,(H_{\mathrm{I}}\,mod\,C_P)\begin{cases} = 0, & H_I = H_I \\ > 0, & H_I = H_I - (H_{\mathrm{I}}\,mod\,R_P) \end{cases} \tag{2}$$

$$W_P = \frac{W_I}{C_P} \tag{3}$$

$$H_P = \frac{H_I}{R_P} \tag{4}$$

After getting (W_P) and (H_P), the next step is to prepare black square (B_S) where all RGB value is 0 with width and height like (W_P) and (H_P). The black square is arranged in random position of (C_P, R_P) on (I_I) as much as (M_P) to cover image in size (W_P) x (H_P). In the previous research, we use double check location function to avoid the same location of (B_S). In this research, we try to use different way to place (B_S) on the (I_I). To place (B_S), we need to calculate how many pieces will be in this (I_{JP}), the calculation got from (C_P) x (R_P). From this number, we produce some number from 1…(C_P x R_P), moreover, we use this group of number to determine location of (B_S). To produce new location of (B_S), we need shuffle the order of this number becoming random position then get the first number from the shuffle's result (S_R). From this number we can use it to calculate (C_{PM}, R_{PM}) "Eq. (5)".

$$if\,(S_{\mathrm{R}}\,rem\,C_P)\begin{cases} = 0, & \begin{matrix} C_{PM} = \lfloor S_R \setminus C_P \rfloor \\ R_{PM} = C_P \end{matrix} \\ > 0, & \begin{matrix} C_{PM} = \lfloor S_R \setminus C_P \rfloor + 1 \\ R_{PM} = S_R\,rem\,C_P \end{matrix} \end{cases} \tag{5}$$

After getting position of (B_S), we remove this number from the group. By using this step, we can repeat the process until it becomes (M_P). The final result of this process is a full image of borderless jigsaw puzzle in RGB type (I_{JP}) with its information of width (W_{JP}) and height (H_{JP}). From this method, we can increase the speed of generation from time1 in the previews research become time2. In this research, we can see it in the "Table 1-3.".

Missing Pieces	Avg Tme1 (seconds)	Avg Time2 (seconds)	Different (%)
25 %	0.114947	0.090565	78.78848513
50 %	0.124699	0.114659	91.94861226
75 %	0.193362	0.144918	74.94647645
99 %	0.478032	0.168951	35.34303143

TABLE 1: Comparison of time consumption of missing pieces generator in 100 pieces.

Missing Pieces	Avg Tme1 (seconds)	Avg Time2 (seconds)	Different (%)
25 %	0.549906	0.507341	92.25958618
50 %	0.917115	0.740216	80.71136117
75 %	1.50405	0.967635	64.3352947
99 %	4.561883	1.200423	64.3352947

TABLE 2: Comparison of time consumption of missing pieces generator in 1000 pieces.

Missing Pieces	Avg Tme1 (seconds)	Avg Time2 (seconds)	Different (%)
25 %	5.078661	4.831655	95.1363952
50 %	9.063756	7.505239	82.80495415
75 %	16.292524	10.109894	62.0523499
99 %	51.176405	12.664193	24.74615597

TABLE 3: Comparison of time consumption of missing pieces generator in 10000 pieces.

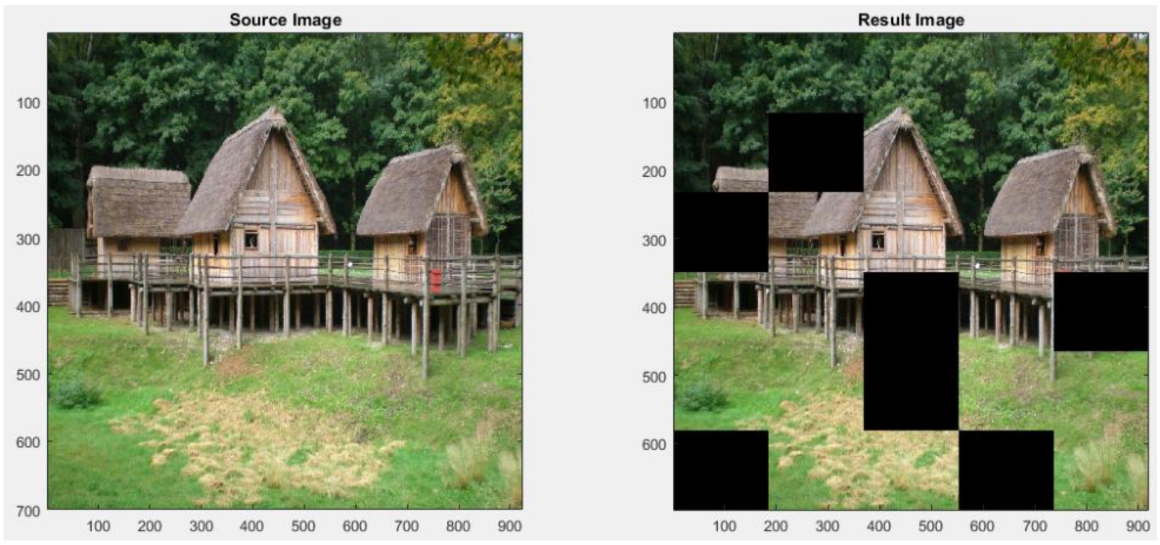

FIGURE 2: Result of Borderless Jigsaw Puzzle Image Formation.

3. CONVERTING AND DETECTING THE COLOR

To convert the color in (I_{JP}) except the black color, we add all RGB value in each pixel. If the result is more than 0 (black color), we change the value becomes 255 (white color) (I_{JPW}) “Eq. (5)”. To fill image regions and holes inside object with white color, we used morphological reconstruction [19]. To make it easier to detect the color, we convert (I_{JPW}) become grayscale image (I_{JPG}). The process makes it has two color, black and white. We can reverse the color From (I_{JPG}) we need to reversed color (I_{JPR}) “Eq. (6)”. One of them will be used to find the smallest square (SQ_N) with (W_{SQ}) x (H_{SQ}), it depends on the smallest number of black color.

$$\sum_{x=1}^{(x+W_{\mathrm{JP}}-1)} \sum_{y=1}^{(y+H_{\mathrm{JP}}-1)} if\ sum\,(I_{\mathrm{JP}}(x,y)) \begin{cases} = 0, & I_{\mathrm{JPW}}(x,y) = 0 \\ > 0, & I_{\mathrm{JPW}}(x,y) = 255 \end{cases} \tag{6}$$

$$\sum_{x=1}^{(x+W_{\mathrm{JPG}}-1)} \sum_{y=1}^{(y+H_{\mathrm{JPG}}-1)} if\ sum\,(I_{\mathrm{JPG}}(x,y)) \begin{cases} = 0, & I_{\mathrm{JPR}}(x,y) = 255 \\ = 255, & I_{\mathrm{JPR}}(x,y) = 0 \end{cases} \tag{7}$$

The last step in this process is detecting edge of black color in (I_{JPG}) by using [20], if in certain location, it contain only black color, it can be saved to [B_{IJPG} (x, y)] "Eq. (7)". This process also applies to (I_{JPR}) which can be saved to [B_{IJPR} (x, y)] "Eq. (8)" as well.

$$\sum_{x=1}^{(x+W_{\mathrm{JPG}}-1)} \sum_{y=1}^{(y+H_{\mathrm{JPG}}-1)} if\ sum\,(I_{\mathrm{JPG}}(x,y)) = 0, B_{\mathrm{IJPG}}(x,y) = 0 \tag{8}$$

$$\sum_{x=1}^{(x+W_{\mathrm{JPR}}-1)} \sum_{y=1}^{(y+H_{\mathrm{JPR}}-1)} if\ sum\,(I_{\mathrm{JPR}}(x,y)) = 0, B_{\mathrm{IJPR}}(x,y) = 0 \tag{9}$$

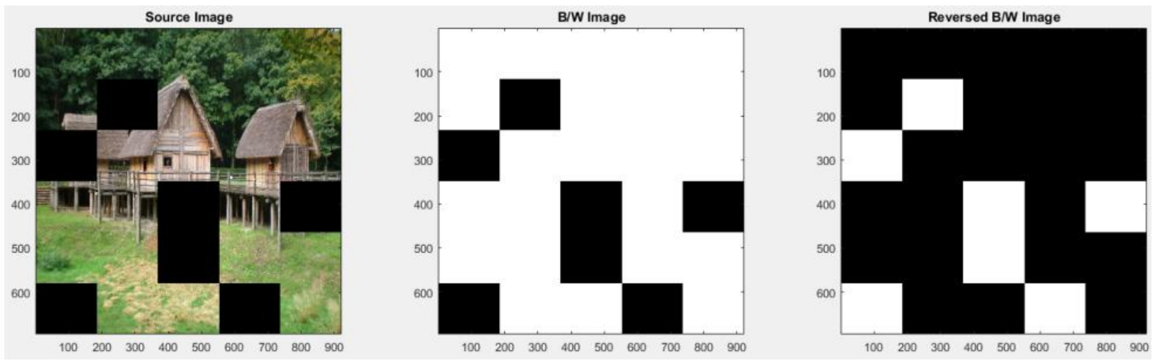

FIGURE 3: Jigsaw Puzzle pieces with border and pieces without border.

4. ASSEMBLING THE SQUARE

The black square is the missing piece in the jigsaw puzzle. From the information in [B_{IJPG} (x, y)] and [B_{IJPR} (x, y)], we try to assemble the square. The first step in this process, we start to count the smallest black color between [B_{IJPG} (x, y)] and [B_{IJPR} (x, y)], if the same number still appears in black color, we can choose one of them. But, if one of them has smallest black square, we choose it as the input for next process. Then, Assemble the vertical line from the same y absis position (L_N) from the input before. When it is found the space between one position with other position in y absis position, it can be assumed as new vertical line (L_{N+1}) "Eq. (9)" and "Eq. (10)".

$$\sum_{counter=1}^{size(B_{\mathrm{IJPG}}(x,y))} if\ B_{\mathrm{IJPG}}(x_{\mathrm{counter}}, y_{\mathrm{counter+1}}) \begin{cases} = 0, & L_{\mathrm{N}} = L_{\mathrm{N}} + 1 \\ > 0, & \begin{matrix} N = N + 1 \\ L_{\mathrm{N}} = L_{\mathrm{N}} + 1 \end{matrix} \end{cases} \tag{10}$$

$$\sum_{counter=1}^{size(B_{\mathrm{IJPR}}(x,y))} if\ B_{\mathrm{IJPR}}(x_{\mathrm{counter}}, y_{\mathrm{counter+1}}) \begin{cases} = 0, & L_{\mathrm{N}} = L_{\mathrm{N}} + 1 \\ > 0, & \begin{matrix} N = N + 1 \\ L_{\mathrm{N}} = L_{\mathrm{N}} + 1 \end{matrix} \end{cases} \tag{11}$$

In the next step of this process is combine all vertical lines (L_N) with the same height (H_{LN}) and start position in y (Y_{LN}) becomes one square (SQ_N), if there is different height on it or different start in position, it is assumed as the new square (SQ_{N+1}) "Eq. (11)".

$$\sum_{counter=1}^{size(L_{\mathrm{N}})} \begin{cases} if\ (H_{L_{\mathrm{N(counter)}}}) \begin{cases} = (H_{L_{\mathrm{N(counter+1)}}}), & \begin{matrix} SQ_{\mathrm{N}} = SQ_{\mathrm{N}} + 1 \\ H_{\mathrm{SQ_N}} = H_{\mathrm{L_{N(counter)}}}, W_{\mathrm{SQ_N}} = W_{\mathrm{L_{N(counter)}}} \\ N = N + 1 \end{matrix} \\ \neq (H_{L_{\mathrm{N(counter+1)}}}), & \begin{matrix} SQ_{\mathrm{N}} = SQ_{\mathrm{N}} + 1 \\ H_{\mathrm{SQ_N}} = H_{\mathrm{L_{N(counter)}}}, W_{\mathrm{SQ_N}} = W_{\mathrm{L_{N(counter)}}} \end{matrix} \end{cases} \\ if\ (Y_{L_{\mathrm{N(counter)}}}) \begin{cases} = (Y_{L_{\mathrm{N(counter+1)}}}), & \begin{matrix} SQ_{\mathrm{N}} = SQ_{\mathrm{N}} + 1 \\ H_{\mathrm{SQ_N}} = H_{\mathrm{L_{N(counter)}}}, W_{\mathrm{SQ_N}} = W_{\mathrm{L_{N(counter)}}} \\ N = N + 1 \end{matrix} \\ \neq (Y_{L_{\mathrm{N(counter+1)}}}), & \begin{matrix} SQ_{\mathrm{N}} = SQ_{\mathrm{N}} + 1 \\ H_{\mathrm{SQ_N}} = H_{\mathrm{L_{N(counter)}}}, W_{\mathrm{SQ_N}} = W_{\mathrm{L_{N(counter)}}} \end{matrix} \end{cases} \end{cases} \tag{12}$$

Moreover, in that process we can identify width (W_{SQ}) and height (H_{SQ}) of each square (SQ_N) from "Eq. (11)". The last step of this process is to find the smallest width (WS_{SQ}) "Eq. (12)" and height (HS_{SQ}) "Eq. (13)" for each square (SQ_N). This is how BALSEM work.

$$WS_{\mathrm{SQ}} = min(W_{\mathrm{SQ}}) \tag{13}$$

$$HS_{\mathrm{SQ}} = min(H_{\mathrm{SQ}}) \tag{14}$$

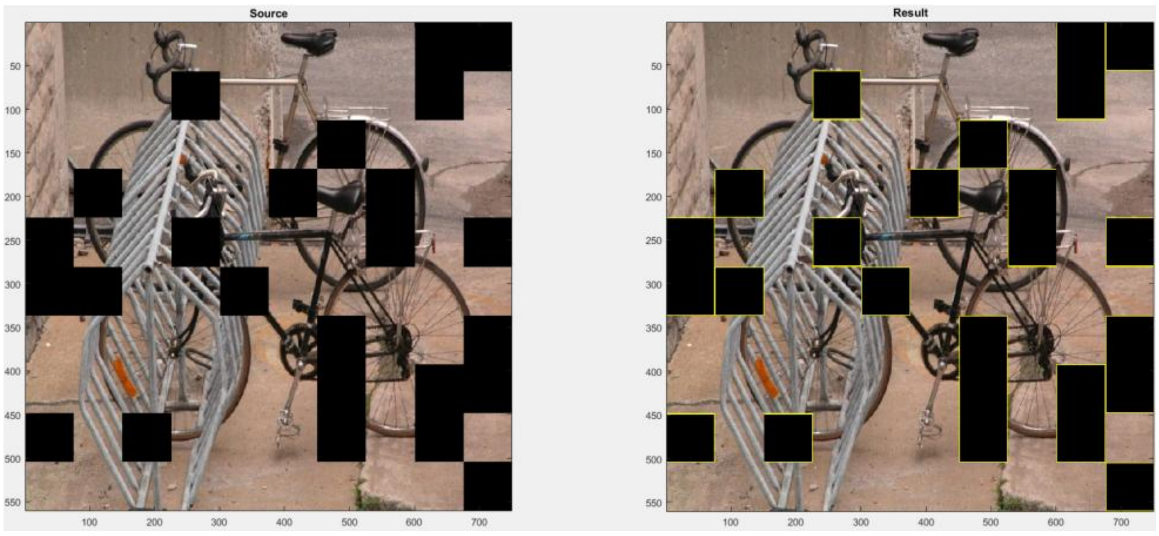

FIGURE 4: Square Assemble and Square Separator.

5. MERGE AND LABELING

The final stage is to find and identify missing piece. From the information of width (WS_{SQ}) and height (HS_{SQ}) of square (SQ_N), we will try to get how many squares in columns (C_{JP}) and rows (R_{JP}). To do this we divide (W_{JP}) with (W_{SQ}) for (C_{JP}) "Eq. (14)" and divide (H_{JP}) with (H_{SQ}) for (R_{JP}) "Eq. (15)" and arrange the all squares (SQ_N) in each (C_{JP}) and (R_{JP}). Starting this process, we also need to identify various colors inside the square. If there is only have 0 value (black color), it means that this is missing piece and the other is not the missing piece. Besides, we put some mark inside that square, (M) for a missing piece, and (P) for not a missing piece as well as the position of that square (SQ_N) in format (column, row). In this research we use three type image size that is 756x560, 980x644 and 1848x1400. In each size we use 20 image and all of this image is in JPG format. The pieces that we use for this research is from 100 pieces to 10000 pieces and either for the missing pieces, we use 25%, 50%, 75% and 99% of missing pieces.

$$C_{JP} = \frac{W_{JP}}{W_{SQ}} \tag{15}$$

$$R_{JP} = \frac{H_{JP}}{H_{SQ}} \tag{16}$$

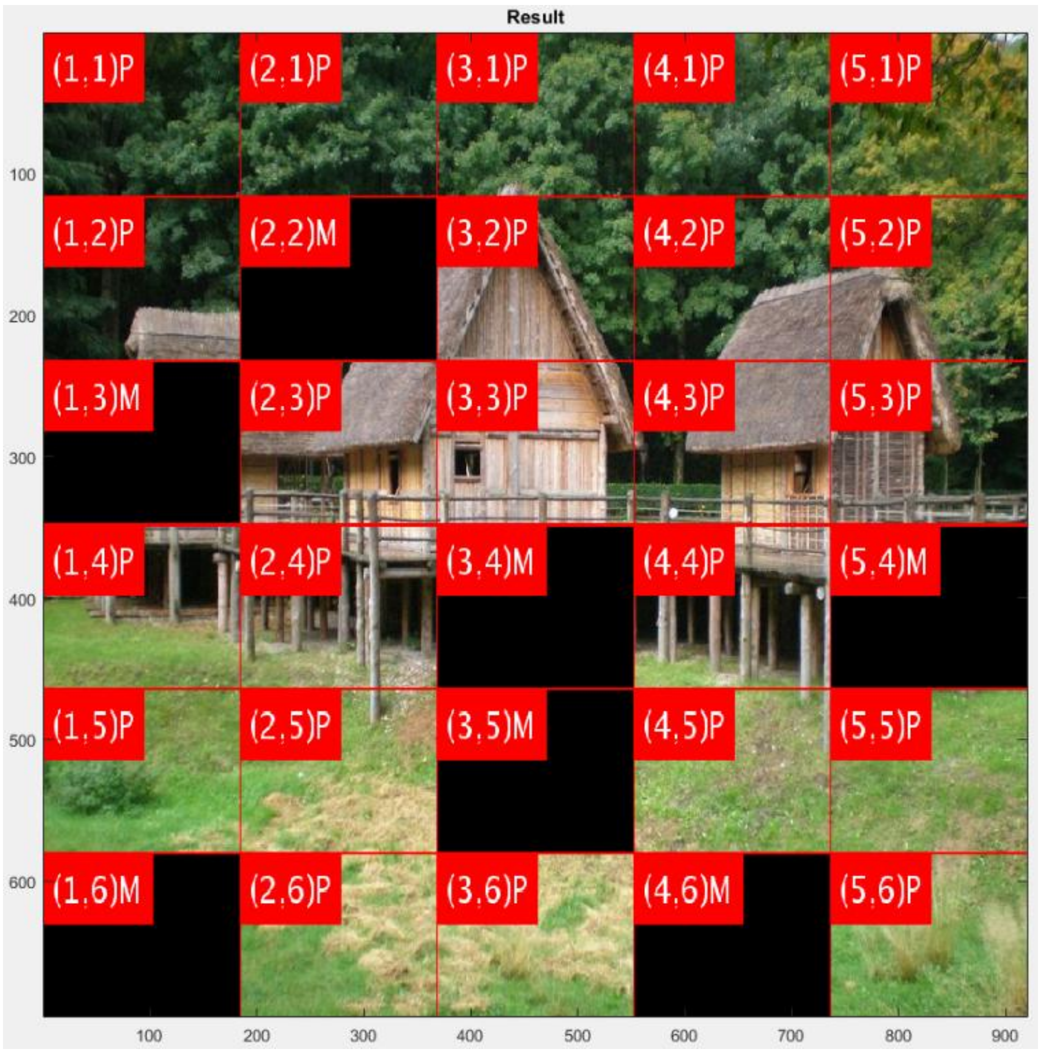

FIGURE 5: Last Result of Labeling and Detection.

6. CONCLUSION

In this research, we present a method to detect missing pieces in the Borderless Square Jigsaw Puzzle. This method we name it BALSEM (Blob Analysis with Line SEarch based on a coluMn). In addition, this method demonstrates how to detect the black color of missing pieces in the borderless square jigsaw puzzle. We can extract the square that merges to another square into separate square (in step assembling the square) as well. As we can see at the "Table 4.", this research results that when (I_{JP}) has large size (W_{JP}) x (H_{JP}) as well as having a lot of missing pieces (M_P) about 50%, it needs more than 580 seconds to finish finding and labeling the missing pieces. From the table above, we can see that the process takes times when the missing pieces are at 50% of the total pieces, on the contrary, when the missing pieces less than 50% or more than 50% the time to finish the process is more faster. In the simulation, the size of the image influences the additional time of finding the missing pieces. Furthermore, the number of missing pieces is able to influence the time reduction in finding the missing pieces as well. The wider of the image, the longer of finding the missing pieces. Then, the more of the number of the missing pieces, the faster of finding the missing pieces.

Image Size	Total Pieces	Missing Pieces	Avg Time (seconds)
756 x 560	100	25 %	169.362292
756 x 560	100	50 %	498.887405
756 x 560	100	75 %	288.041643
756 x 560	100	99 %	14.368075
980 x 644	1000	25 %	169.949923
980 x 644	1000	50 %	525.054162
980 x 644	1000	75 %	244.056859
980 x 644	1000	99 %	10.82156
1848 x 1400	10000	25 %	211.291794
1848 x 1400	10000	50 %	580.612713
1848 x 1400	10000	75 %	284.595062
1848 x 1400	10000	99 %	16.884004

TABLE 4: Comparison of time consumption for finding and labeling missing pieces.

The obstacle in this research appears because the method we use only enable to scan and compare each pixel with another pixel and it is used only one process between (I_{JPG}) and (I_{JPR}), in which the less black pieces are. The future work will be useful for using some optimization method to increase the speed for finding missing piece in the Borderless square jigsaw puzzle and also, the researchers are trying to add the noise that is the rotation of the puzzle. If it is done by adding the rotation in it, and the system is still able to detect the missing pieces well, then we can apply the real time detection by using camera in it

7. REFERENCES

[1] S. Novianto, K. Penantcha, and F. Luo, "Labeling and Finding Missing Pieces of Jigsaw Puzzle," in *ISemantic - 2016*, 2016.

[2] P. J. Chiang, M. J. Tseng, Z. S. He, and C. H. Li, "Automated counting of bacterial colonies by image analysis," *J. Microbiol. Methods*, vol. 108, pp. 74–82, 2015.

[3] J. E. Arco, J. M. Górriz, J. Ramírez, I. Álvarez, and C. G. Puntonet, "Digital image analysis for automatic enumeration of malaria parasites using morphological operations," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3041–3047, 2015.

[4] E. Imani, H.-R. Pourreza, and T. Banaee, "Fully automated diabetic retinopathy screening using morphological component analysis," *Comput. Med. Imaging Graph.*, vol. 43, pp. 78–88, 2015.

[5] S. Nazlibilek, D. Karacor, T. Ercan, M. H. Sazli, O. Kalender, and Y. Ege, "Automatic segmentation, counting, size determination and classification of white blood cells," *Meas. J. Int. Meas. Confed.*, vol. 55, pp. 58–65, 2014.

[6] N. Batool and R. Chellappa, "Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints," *Pattern Recognit.*, vol. 48, no. 3, pp. 642–658, 2015.

[7] M. Gori, M. Maggini, S. Marinai, J. Q. Sheng, and G. Soda, "Edge-backpropagation for noisy logo recognition," *Pattern Recognit.*, vol. 36, no. 1, pp. 103–110, 2003.

[8] D. Murkherjee, A. Pal, E. Sarma, and D. D. Majumder, "Water Quality Analysis: A Pattern Recognition Approach," *Pattern Recognit.*, vol. 28, no. 2, pp. 269–281, 1995.

[9] S. A. Sirohey and R. Azriel, "Eye detection in a face image using linear and nonlinear filters," *Pattern Recognit.*, vol. 34, no. 7, pp. 1367–1391, 2001.

[10] Q. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudo ridges," *Pattern Recognit.*, vol. 37, no. 11, pp. 2233–2243, 2004.

[11] P. L. Rosin, "A simple method for detecting salient regions," *Pattern Recognit.*, vol. 42,

no. 11, pp. 2363–2371, 2009.

[12] Y. D. Chethan, H. V. Ravindra, Y. T. gowda, and S. Bharath Kumar, “Machine Vision for Tool Status Monitoring in Turning Inconel 718 using Blob Analysis,” *Mater. Today Proc.*, vol. 2, no. 4–5, pp. 1841–1848, 2015.

[13] S. Mitra and B. Uma Shankar, “Medical image analysis for cancer management in natural computing framework,” *Inf. Sci. (Ny).*, vol. 306, pp. 111–131, 2015.

[14] E. S. da Silva and H. Pedrini, “Connected-component labeling based on hypercubes for memory constrained scenarios,” *Expert Syst. Appl.*, vol. 61, pp. 272–281, 2016.

[15] H. Li, Y. Zheng, S. Zhang, and J. Cheng, “Solving a special type of jigsaw puzzles: Banknote reconstruction from a large number of fragments,” *IEEE Trans. Multimed.*, vol. 16, no. 2, pp. 571–578, 2014.

[16] M. Makridis and N. Papamarkos, “A new technique for solving puzzles.,” *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 40, no. 3, pp. 789–97, Jun. 2010.

[17] J. T. Tsai and P. Y. Chou, “Solving Japanese puzzles by genetic algorithms,” *Proc. - Int. Conf. Mach. Learn. Cybern.*, vol. 2, pp. 785–788, 2011.

[18] G. Paikin and A. Tal, “Solving multiple square jigsaw puzzles with missing pieces,” *Comput. Vis. Pattern Recognit. (CVPR), 2015 IEEE Conf.*, pp. 4832–4839, 2015.

[19] S. L. E. Gonzalez, R. C., R. E. Woods, *Digital Image Processing Using MATLAB*. New Jersey: Pearson Prentice Hall, 2004.

[20] C. Theoharatos, G. Economou, and S. Fotopoulos, “Color edge detection using the minimal spanning tree,” *Pattern Recognit.*, vol. 38, no. 4, pp. 603–606, 2005.