Solving Tangram Puzzles Using Raster-Based Mathematical Morphology

Fernanda Miyuki Yamada*, João Paulo Gois† and Harlen Costa Batagelo‡
Center for Mathematics, Computing and Cognition
Federal University of ABC, Santo Andre, Brazil
Email: *fernandamyamada1@gmail.com, †joao.gois@ufabc.edu.br, ‡harlen.batagelo@ufabc.edu.br

***Abstract*—The Tangram is a dissection puzzle composed of polygonal pieces which can be combined to form different patterns. Solving the Tangram is a two-dimensional irregular shape packing problem known to be NP-hard. Tangram patterns may be composed of multiple connected components, and assembling them may require the reflection transformation and unconstrained rotations of the pieces. In this work, we propose a novel approach for the automatic solution of the Tangram based on a raster representation of the puzzle. In order to adapt the geometrical techniques that are applied to the prevention of piece overlapping and the reduction of space between pieces, we use morphological operators and representations commonly used in the discrete domain such as the dilation operator, the distance transform and the morphological skeletonization. We investigate the effects of the raster representation in the puzzle assembly process and verify the effectiveness of the proposed method in solving different Tangram puzzles.**

I. Introduction

The Tangram is a geometric puzzle composed of seven polygonal pieces: a square, a parallelogram, and five triangles of different sizes. The main objective of the Tangram is to rearrange the seven pieces using rigid body transformations to fit them into a given pattern composed of a single or multiple connected planar regions [1]. All pieces must be used and cannot overlap. In Figure 1, the pieces are shown in their initial square configuration. The constraint of seven pieces may be relaxed, and multiple sets of Tangram can be combined to allow for the assembly of more elaborate patterns.

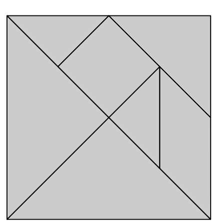

Fig. 1: Tangram pieces in the initial configuration.

Computationally, the Tangram relates to a more general class of combinatorial NP-hard problems such as Cutting and Packing (C&P) problems [2]. The geometrical techniques that are applied in C&P problems to identify and prevent overlaps between shapes can be easily adapted to deal with the same task in Tangram puzzles. The irregular C&P problem consists of a task involving the placement of irregularly-shaped pieces into a container while optimizing an objective function, such as minimizing waste of area. The pieces must not overlap and must stay within the limits of the container [2]. Dissection puzzles are particular cases of the irregular C&P problem in which the number of pieces is fixed and, at the end of the pieces placement process, there is no space left in a container of limited size [3].

It is common for C&P problems to restrict transformations to translations and rotations constrained to multiples of 180°, 90° or 45°, and make use of a single rectangular container. In general, the Tangram assembly process requires irregularly-shaped containers, unconstrained rotations of the pieces, and the additional reflection transformation for the parallelogram.

To better differentiate the complexity of Tangram puzzles, we define *simple Tangram puzzles* as puzzles that can be fully characterized by a set of translations and rotations constrained to multiples of 45° to form a pattern composed of a single connected component. In contrast, we define *complex Tangram puzzles* as the ones which present at least one of the following characteristics: (1) are composed of multiple connected components; (2) contain holes within the puzzle area; (3) require unconstrained rotations for the pieces; (4) require the reflection transformation of the parallelogram. Figure 2 exemplifies complex Tangram puzzles with different attributes.

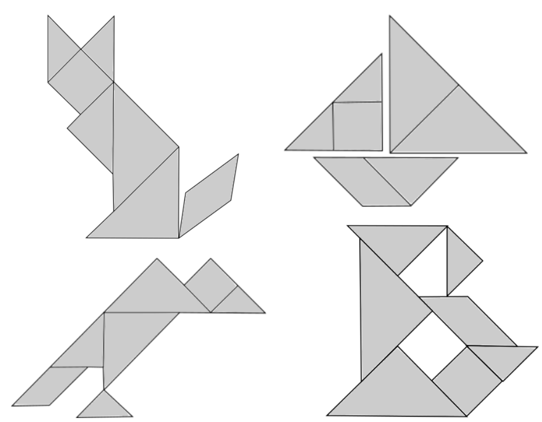

Fig. 2: Examples of complex Tangram puzzles. The cat's tail (top left) is a parallelogram rotated by an angle that is not a multiple of 45°. The boat (top right) is composed of multiple connected components. The bird (bottom left) uses a reflected parallelogram, and the letter B (bottom right) contains holes.

In this paper, we propose a novel approach for the automatic solution of Tangram puzzles. Our method takes advantage of operators and representations of raster-based mathematical morphology to allow for fast placements of the pieces and support for complex Tangram puzzles. During assembly, the puzzle area and pieces are represented as raster masks, which can be combined using bit blit operations to support patterns with holes and patterns composed of multiple connected components. The geometrical techniques that are commonly applied in the solution of C&P problems in vector format are adapted to the raster format. For instance, the computation of no-fit polygons and collision-free areas to avoid overlaps between pieces and the boundaries of the puzzle area are performed using raster-based morphological dilations. This allows using Tangram patterns and pieces described solely by binary raster masks. Similarly, we propose a new placement position strategy based on minimizing a placement cost function that uses the distance transform computed in the empty region. The evaluation of the cost function at the endpoints of the morphological skeleton of the collision-free area quickly determines the positions of potentially minimum cost. We show that this strategy competes well with existing placement strategies in terms of efficiency, and outperforms them in the number of puzzles solved. Therefore, this paper brings the following contributions: (1) an automatic Tangram puzzle solver which can use puzzle patterns and pieces described solely in raster format; (2) a new distance transform placement position strategy that is capable of solving a more extensive variety of puzzles when compared to existing placement strategies used in raster-based C&P techniques, while maintaining efficiency.

The vector-to-raster conversion is accompanied with information loss because the boundaries of the polygons often do not coincide with the boundaries of raster cells. However, we found that this loss in the representation accuracy during assembly does not have a significant impact on the puzzles since it is enough for the Tangram to create patterns that visually resemble symbols and objects. Even if the pieces are not perfectly arranged, the solution may be accepted as correct if the resulting pattern resembles the desired symbol or object.

In Section II, we present a synthesis of the literature of computational Tangram solving techniques and a literature review of the raster representations that are applied to the solution of C&P problems. Our method is presented in Section III. In Section IV, we present and discuss the results of the application of our method on different Tangram puzzles. Finally, in Section V, we present our final considerations and future works.

II. Related work

While the literature about irregular C&P problems is vast, there are only a few works devoted to methods for the automatic solution of Tangram puzzles [4]. Deutsch & Hayes presents a method for solving Tangram puzzles using heuristic programming [5]. Oflazer follows a connexionist approach and solves Tangram puzzles by representing the placement and orientation of the pieces as a non-restricted Boltzmann machine [6]. Among the methods dedicated to jigsaw [7] and edge-matching [8] puzzles, the method by Bartoněk [9] based on genetic algorithms includes an extension to dissection puzzles such as the Tangram. More recently, Kovalsky et al. proposed a method based on the solution of systems of polynomial equations derived from the pieces of the puzzle. This method can be used to solve simple Tangram puzzles [10]. All these approaches show limitations concerning the orientation of pieces, usually limiting the angles of rotations to multiples of 45^o. Also, they cannot be applied to patterns with multiple connected regions. Only the methods by Deutsch & Hayes [5] and Oflazer [6] allow the reflection transformation for the parallelogram, and only the method by Oflazer solves patterns with holes.

Besides the methods dedicated to the Tangram and the methods that include extensions to the Tangram, traditional methods applied to C&P problems were proved to be efficient to avoid overlaps among polygons using the concept of no-fit polygons, inner-fit polygons and collision-free areas [2], [3]. In the following, we focus on works using a discrete representation for the shapes and container.

Toledo et al. [11] propose the dotted-board model, in which the container is represented as a grid, and the no-fit polygon is transformed into a no-fit raster represented by a binary matrix. Binary variables are associated with a piece type and a grid dot, where the pieces can be positioned inside the container [12].

MirHassani & Jalaeian Bashirzadeh [13] present a greedy randomized adaptive search metaheuristic based on the no-fit raster and use the constraints of the dotted-board model defined by Toledo et al. [11] to guarantee that the shapes do not overlap. This method takes up to 300 seconds to solve simple problems. Mundim et al. [2] point out that this time could be reduced considerably by using directly the information provided by the no-fit raster.

Mundim et al. [14] present the no-fit raster and the inner-fit raster for free form pieces. The no-fit polygon and the inner fit polygon are converted to binary masks. The quality of the solutions obtained for irregular C&P problems using the raster format depends on the resolution of the discretization used to generate the raster representations of the polygons.

Rodrigues & Toledo [12] propose a clique covering mixed-integer programming model for the irregular strip packing problem based on the dotted-board model proposed by Toledo et al. [11] in which the board is represented by a grid. The authors state that the proposed model obtained better performance than the dotted-board model for most instances and solved larger instances to optimality in comparison to other works of the literature.

Finally, Mundim et al. [2] propose a heuristic to solve the two-dimensional version of the C&P problems with irregular pieces and limited-size containers. Also, the authors used no-fit raster and inner-fit raster concepts to prevent overlaps between pieces. The computational experiments show that the proposed heuristic improved on the best solutions available in the literature for three problems: the placement, the knapsack, and the cutting-stock problems. The authors compare the proposed method concerning the automatic solution of placement and maximization problems with the techniques proposed by Valle et al. [15], Dalalah et al. [16], Fischetti & Luzzi [17], Alvarez-Valdes et al. [18], Gomes & Oliveira [19]. Considering the current scenario of the usage of discrete representation in computational methods for the automatic solution of the C&P problems, the authors do not mention the application of morphological operations in the discrete representation as an approach to detect and prevent the occurrence of overlaps between shapes, as well as to reduce the distance between shapes. Therefore, we consider that our method brings some contribution concerning the use of discrete representation in general two-dimensional C&P problems and presents a new approach for the prevention of overlaps between shapes.

III. The Method

Our method starts with the inputs necessary to execute the assembly process of the Tangram pattern: a Tangram pattern (detailed in Section III-A) and a placement position strategy (Section III-B). As a preprocessing step, the method generates a list of all the possible configurations that each piece can assume considering orientations and reflection transformation. After that, the method proceeds to the placement procedure (Section III-C) responsible for positioning the pieces inside the puzzle area. As each piece is inserted into place, the method performs a validation process (Section III-D) responsible for verifying whether an obtained intermediate solution of the puzzle allows the positioning of the remaining pieces.

A. Tangram puzzle representation

The desired Tangram pattern is represented as a rectangular binary mask which will be referred to as *pattern mask*. At the beginning of the assembly process, we consider that, in the pattern mask, the empty puzzle area is represented by black pixels (0), and the area outside the puzzle region is represented by white pixels (1). During the placement process, if a piece is placed in a black area, all pixels corresponding to the raster conversion of that piece turn white in the pattern mask. Thus, after applying our method in the Tangram puzzle, it is expected that all of the pixels of the pattern mask (or at least most of them) will be white. This representation can easily describe complex Tangram patterns by assigning the regions with holes and spaces between closed contour regions to white pixels. Figure 3 shows examples of binary masks corresponding to complex Tangram patterns.

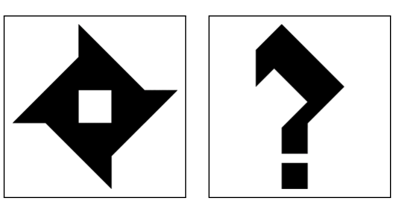

Fig. 3: Examples of complex Tangram patterns. Pattern with a hole (left), and with multiple connected components (right).

Each Tangram piece is represented as a list of vertices in a local coordinate system where the origin is at the piece center. Each piece contains the following attributes: translation, angle of rotation, and a flag indicating whether the piece is reflected. Rotation and reflection are performed locally. The translation is performed with respect to the coordinate system of the pattern mask. At the end of the assembly process, these attributes correspond to the set of transformations required to produce a feasible solution for the Tangram puzzle.

B. Placement position strategies

The placement position strategies aim to guide the positioning of the Tangram pieces according to a pattern that avoids placing the pieces at random inside the puzzle area. We consider the placement position strategies described by Mundim et al. [2] that sweeps the puzzle area based on the patterns illustrated in Figure 4: Bottom Left (BL), Left Bottom (LB), Horizontal Zig-Zag (HZZ), Vertical Zig-Zag (VZZ), Spiral (S), Anti Spiral (S^{-1}).

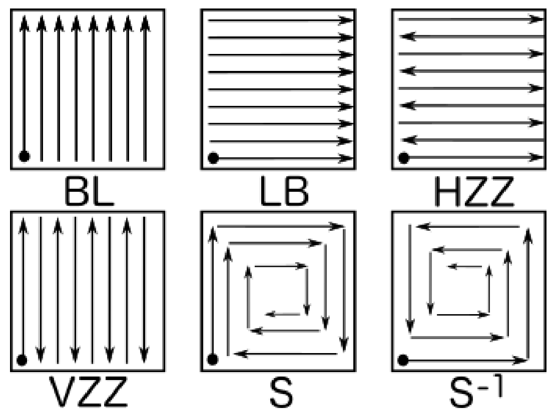

Fig. 4: Paths generated by each placement position strategy.

In our method, the pattern obtained from each placement position strategy is converted to a matrix that has the same size as the pattern mask. Each entry of the matrix represents a placement cost. Figure 5 illustrates the matrices generated from each placement position strategy using intensity values to depict the cost. The role of this cost in the placement procedure is detailed in Section III-C.

We also introduce a new placement position strategy based on the distance transform. The distance transform maps each pixel of the empty area of the pattern mask to its shortest Euclidean distance with respect to the outside puzzle area. We consider the distance values as placement costs in order to place the pieces as close as possible to the outside area of the puzzle while avoiding overlapping. Figure 6 exemplifies a matrix generated from the proposed approach based on the distance transform. Similar to the matrix obtained from the placement position strategies [2], the values from the distance transform are normalized to be between 0 and 1.

C. Pieces placement procedure

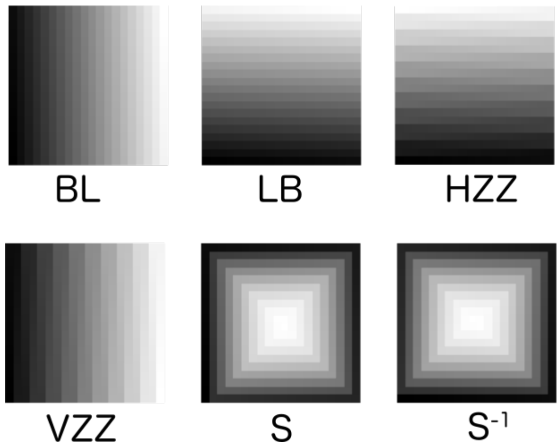

Fig. 5: Matrices of placement position strategies with values in grayscale, from black (lowest cost), to white (highest cost).

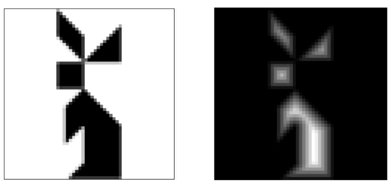

Fig. 6: A pattern mask (left) and the corresponding distance transform (right) with distances shown in grayscale.

Algorithm 1:

```
1  Set current piece;
2  while all pieces not placed do
3      if list of possible configurations of the current piece
         is not empty then
4          Get next configuration of current piece;
5          Find feasible positions of current piece;
6          if feasible position not found then
7              continue;
8          end
9          Execute placement on the lowest cost position;
10         if following pieces can be placed in the empty
             regions then
11             Set the following piece as the current piece;
12             continue;
13         end
14         else
15             Remove the current piece;
16             continue;
17         end
18     end
19     else if there are pieces in the puzzle area then
20         Reset the list of possible configurations of the
             current piece;
21         Remove the previous piece;
22         Set previous piece as the current piece;
23     end
24     else
25         return Empty Solution;
26     end
27 end
28 return Solution;
```

Our method applies the largest-first heuristic, in which the Tangram pieces are sorted according to their area, and the largest pieces are placed first [20]. Algorithm 1 presents the pseudocode of the pieces placement procedure.

At the beginning of the pieces placement procedure, the method selects a Tangram piece following the largest-first heuristic (Line 1). Then, it selects the possible configurations of the current piece one at a time to execute the pieces placement procedure (Line 4). The next step consists of finding a position for the current piece in the current configuration inside the puzzle area (Line 5).

In order to determine the feasible positions, the raster collision-free area is calculated. The collision-free area represents all possible translations for an item to be placed. Since the pattern mask contains the union of all the placement obstacles (placed pieces, holes and exterior areas), to obtain the collision-free area, the pattern mask is dilated using the reflected configuration mask of the current piece as the structuring element of the dilation.

To reduce the number of positions to be considered, the method computes the endpoints of the morphological skeleton of the collision-free area. These correspond to concave corners of the collision-free area, which provide a better interaction between the piece corners and the corners of the non-empty areas (limits of the Tangram pattern and pieces placed inside the puzzle area). Some corners may be missed by this method, but this is traded off by efficiency.

If any collision-free area could not be found (Lines 6, 7 and 8), the method returns to the possible configuration selection step, and attempts to get the next possible configuration of the current piece.

Figure 7 (a–e) presents the process for obtaining the feasible positions of a piece. Figure 7 (f–g) illustrates the process of placing the piece at the lowest cost position according to our distance transform placement position strategy. The computation of the cost is discussed in the following.

During the endpoints evaluation process (Line 9), we use a cost function to determine the cost associated with each feasible positioning of the current piece in the current configuration, taking into consideration the puzzle area, and the other pieces that were already placed. Our cost function is defined as:

$$c(\vec{x}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(M(\vec{x})\right)_{i,j}, \tag{1}$$

where c is the cost function, $\vec{x}$ is an array representing a possible placement for the current piece, $M(\vec{x})$ is the placement cost matrix that represents the placement cost associated with $\vec{x}$, n and m are the number of rows and columns of $M(\vec{x})$.

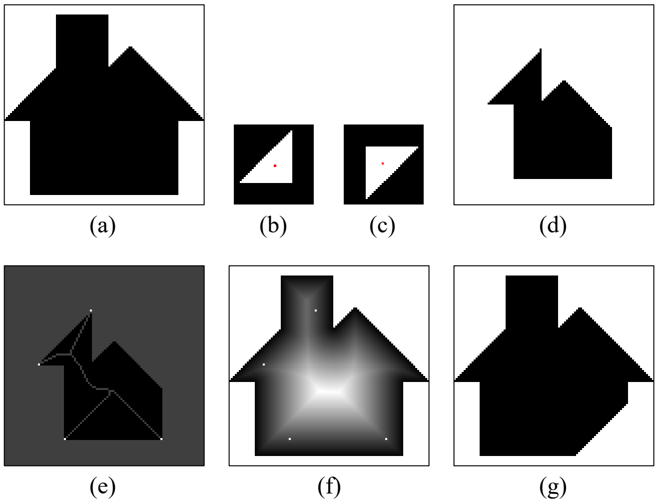

Fig. 7: Computing feasible positions for a piece and placing it at the lowest cost position: (a) Pattern mask; (b) Configuration mask of the current piece (origin marked in red); (c) Reflected piece; (d) Collision-free area obtained by dilating (a) with (c); (e) Endpoints of the morphological skeleton of the collision-free area shown as white dots. For reference, the skeleton and collision-free area shown in gray; (f) Composite image showing the distance transform inside the pattern mask and the endpoints which correspond to feasible positions for (b); (g) Resulting pattern mask after placing (b) at the lowest cost position, which in this case corresponds to the bottom right endpoint in (e-f).

In order to determine the placement cost matrix associated with a feasible position found for the current piece in the current configuration, it is necessary to take into consideration the placement position strategy that was determined at the beginning of the method execution. The process for obtaining the cost function for instances using the placement position strategies described in the work of Mundim et al. [2] is illustrated in Figure 8 using the Left Bottom strategy.

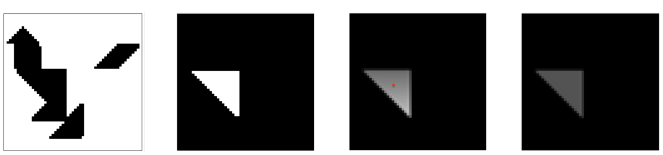

Fig. 8: Process for obtaining the placement cost matrix of the current piece using a placement position strategy. Pattern mask (left), current piece configuration mask (center-left), element-wise multiplication of current piece configuration mask and placement position strategy mask with the centroid represented as a red dot (center-right), and final placement cost matrix (right).

Aiming to obtain the placement cost matrix associated with the current piece positioning, it is necessary to perform an element-wise multiplication between the current piece mask and the matrix generated from the placement position strategy determined at the beginning of the method execution. After that, the current piece centroid is calculated to determine the value of the element located in the current piece center. Finally, the value of this central element is extended over the elements of the piece area in the current piece mask to obtain the placement cost matrix $M(\vec{\boldsymbol{x}})$ that represents the placement cost associated with the possible placement $\vec{\boldsymbol{x}}$.

The process for obtaining the cost function for instances using the distance transform approach proposed in this study is presented in Figure 9.

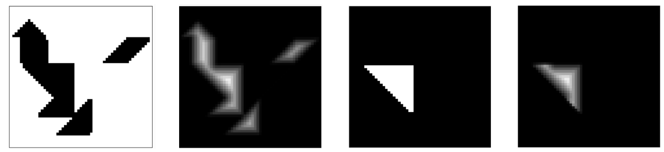

Fig. 9: Process for obtaining the placement cost matrix of the current piece using the distance transform. Pattern mask (left), distance transform (center-left), current piece configuration mask (center-right), and final placement cost matrix (right).

In order to obtain the cost associated with the current piece positioning, it is necessary to calculate the distance transform mask that determines the distance between each element in the puzzle area, and the obstacles represented in the pattern mask, i.e., placed pieces, holes and exterior areas. Then, an element-wise multiplication is executed between the current piece mask, and the distance transform mask to obtain the placement cost matrix $M(\vec{\boldsymbol{x}})$ that represents the placement cost associated with the possible placement $\vec{\boldsymbol{x}}$.

In C&P problems, the objective of the placement position strategies is to define a sequence of positions to be tested one after another aiming to guide the shapes positioning process. In general, the shape placement process is executed by placing the center of the shape on the positions, following the sequence established by the placement position strategy. The process for obtaining the placement cost matrix of the current piece using the mask generated from the placement position strategy gives preference to the positions that come first in the placement position strategy path by assigning them a lower cost value. On the other hand, the distance transform approach aims to approximate the current piece to Tangram pattern borders, as well as other Tangram pieces already placed inside the puzzle area. In this approach, we consider the cost associated with the whole area occupied by the current piece in the puzzle area, not just the center of the piece.

D. Validation

After the current piece placement, the method proceeds to a validation process (Lines 10 to 17 in Algorithm 1), which is responsible for verifying whether the empty regions can accommodate all the subsequent pieces in the largest-first heuristic. In this validation process, the following conditions have to be satisfied: (1) the largest empty region must be greater than the largest piece in number of pixels and (2) the maximum value of the distance transform considering the largest piece must be shorter than the maximum value of the distance transform considering the empty regions. This verification is used in the proposed method to determine whether an intermediate solution is feasible, taking into consideration the following iterations of the proposed method. If the placement validation process considers that the current piece placement does not prevent the placement of the subsequent pieces, then the method proceeds to the placement procedure of the following piece determined by the largest-first heuristic. Otherwise, the method removes the current piece from the puzzle area and returns to the step in which it attempts to get the next possible configuration of the current piece. Figure 10 shows examples in which the validation verified that the following pieces would not be able to fit inside the empty regions. In both cases, the puzzle considered is the initial configuration of the Tangram puzzle, and the pieces that were still not placed inside the puzzle area were the two small triangles and the medium triangle.

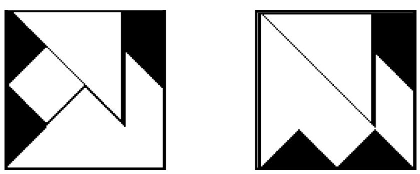

Fig. 10: Cases in which the placement validation considered that the following pieces would not fit in the empty areas.

Furthermore, if the method gets to the point in which all the possible configurations of the current piece were already considered, and none of them produced a feasible placement for the current piece, the method verifies if there is at least one piece placed inside the puzzle area (Lines 19 to 26 in Algorithm 1), and if there are pieces placed inside the puzzle area, the method resets the list of possible configurations of the current piece, removes the previous piece considering the largest-first heuristic, and attempts to find a new placement for that piece considering another possible configuration. Otherwise, if there is not any piece left in the puzzle area, the method finishes its execution with none of the pieces placed inside the puzzle area, indicating that it was not possible to find a solution for the given Tangram pattern. Finally, the method considers that the Tangram puzzle was successfully assembled if the heuristic places all the Tangram pieces inside the puzzle area (Line 28 in Algorithm 1).

IV. RESULTS

In our dataset, we considered Tangram puzzles with different features, including patterns with holes, patterns composed by multiple connected components, and patterns in which the pieces are not limited to a discrete set of 45° multiple orientations. Figure 11 presents the Tangram patterns included in our dataset.

Concerning the reflection transformation, it is not possible to easily ascertain whether a Tangram puzzle demands the reflected parallelogram. Even if one of the solutions of a particular Tangram puzzle is known, other solutions for the same puzzle might exist, since different arrangements of pieces can result in the same pattern [21]. Thus, for a single Tangram puzzle, it is possible to find either feasible solutions that contain the reflected parallelogram and solutions that do not contain the reflected parallelogram. Therefore, to test the reflection transformation requirement in the Tangram patterns included in our dataset, we conducted different experiments, including the implementation and the non-implementation of the reflection transformation.

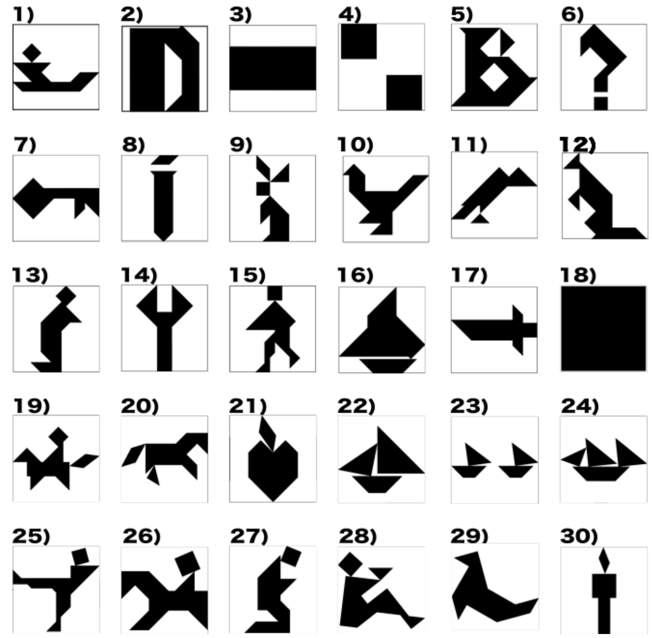

Fig. 11: Tangram patterns that compose our dataset.

We elaborated three different experiments considering: (1) unconstrained rotations and reflection transformation, (2) constrained rotations, and reflection transformation, and (3) constrained rotations and no reflection transformation. The purpose of executing these experiments in succession is to progressively reduce the complexity of the Tangram puzzle solution problem, in terms of the number of configurations a Tangram piece can assume during the Tangram puzzle assembly process. We start testing our proposed method on the most complex scenario, in which it is expected from the algorithm to be able to solve Tangram puzzles with rotations that are not multiples of 45°, and Tangram puzzles that might require the reflection transformation. The number of angles of rotation has a more significant impact on the number of Tangram pieces possible configurations than the reflection transformation, which is just performed on the parallelogram. Finally, the execution of these experiments permits the analysis of the influence of the progressive complexity reduction on the obtained solutions and on the average running time necessary to assemble the Tangram patterns included in our dataset.

The number of Tangram puzzles considered in each experiment is different. In the first executed experiment, in which we consider unconstrained rotations, the Tangram puzzles tested are described by patterns 1 to 30 in Figure 11. In the following experiments, in which we consider constrained rotations, the Tangram puzzles considered are described by patterns 1 to 18 in Figure 11. This difference occurs because patterns 19 to 30 of Figure 11 are composed of rotations that are not limited to multiples of 45^o.

Furthermore, in the first executed experiment, to implement the rotations that are not multiples of 45^o without increasing the solution space of our problem to impracticable proportions, we implemented the rotations in a range from 0^o to 359^o with increments of 1^o. Since the main objective of a Tangram puzzle is to find an arrangement of Tangram pieces that visually reassembles the correspondent Tangram pattern, we consider that the precision of 1^o is sufficient to establish a visual resemblance between them, as well as to maintain a feasible solution space.

The method was implemented in MATLAB with the Image Processing Toolbox. The tests were performed on an Intel Core i5-5250U 1.6 GHz, 4GB of RAM, using a single thread. We considered the running time as a measure of the performance of the method, and establish a time limit of 360s for the method to find a feasible solution for a Tangram puzzle. We assume that 360s is the maximum time a user is willing to wait for a feasible solution for a given Tangram puzzle.

Table I presents an overview of the proposed method execution, taking into consideration the number of solved Tangram puzzles according to each conducted experiment and each placement position strategy applied.

TABLE I: Number of puzzles solved in each experiment.

Placement position strategy	Solved Tangram puzzles in experiment 1	Solved Tangram puzzles in experiment 2	Solved Tangram puzzles in experiment 3
BL	19	8	2
LB	21	11	6
HZZ	20	10	6
VZZ	20	9	2
S	19	8	2
S^{-1}	19	8	2
DT	23	15	7

The obtained results show that: in the first experiment, the method was able to solve 141 Tangram puzzles (79 composed of constrained rotations and 62 composed of unconstrained rotations); in the second experiment, the method was able to solve 69 Tangram puzzles; and in the third experiment, the method was able to solve 27 Tangram puzzles. Since different numbers of Tangram patterns were considered during the execution of the experiments, Table II presents the percentage of Tangram puzzles solved in each experiment and the percentage of Tangram puzzles solved following the placement position strategies presented in the work of Mundim et al. [2] and the percentage of Tangram puzzles solved following the proposed distance transform approach.

Table III presents an overview of the proposed method execution. It considers the average running time the method took to solve the Tangram puzzles according to each conducted experiment and each placement position strategy applied.

TABLE II: Percentage of puzzles solved in each experiment.

Experiment	Solved Tangram puzzles	Solved Tangram puzzles using placment position strategies	Solved Tangram puzzles using the distance transform approach
Exp. 01	67.14%	65.56%	76.67%
Exp. 02	54.76%	50.00%	83.33%
Exp. 03	21.43%	18.52%	38.89%

TABLE III: Average running time to solve a Tangram puzzle in each experiment and their correspondent standard deviation.

Placement position strategy	Experiment 1	Experiment 2	Experiment 3
BL	98.90s (93.85s)	5.75s (7.51s)	15.38s (12.52s)
LB	92.21s (102.22s)	4.14s (5.98s)	4.20s (4.59s)
HZZ	89.15s (104.39s)	2.35s (1.78s)	4.20s (4.55s)
VZZ	90.66s (89.33s)	5.70s (8.54s)	15.90s (13.00s)
S	67.32s (80.63s)	2.23s (1.96s)	1.02s (0.17s)
S^{-1}	68.72s (87.84s)	2.24s (1.97s)	1.01s (0.16s)
DT	76.11s (94.04s)	4.90s (9.62s)	5.29s (7.72s)

The average running time the method took to solve a Tangram puzzle presented a considerable variation among the executed experiments. On average, the first experiment required a higher running time to solve a puzzle when compared to the other two experiments. Therefore, we observe a relation between the complexity concerning the number of configurations that the pieces can assume and the average running time necessary to correctly assemble a Tangram puzzle.

Figure 12 presents examples of obtained solutions to illustrate the precision of the proposed discrete representation taking into consideration the visual resemblance between the Tangram pattern and the corresponding obtained solution.

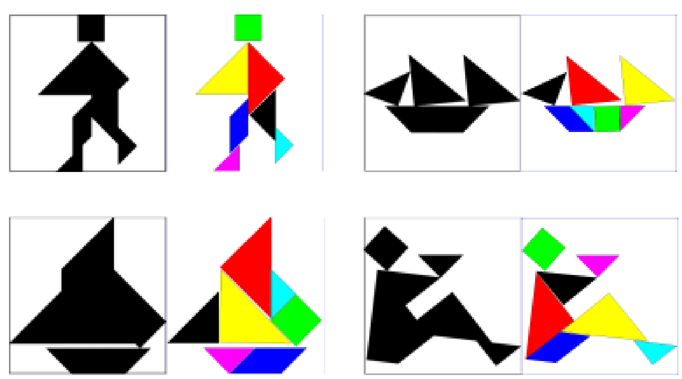

Fig. 12: Examples of solutions found through the execution of our method.

V. Conclusion

In this paper, we proposed a heuristic method for the automatic solution of Tangram puzzles, aiming to solve the main limitations identified in the literature. In order to accomplish our main objective, we propose a raster representation that permits to describe patterns with holes, and composed by multi-connected regions. Besides, concerning the transformations to be performed on the Tangram pieces, our method implements the reflection transformation and permits the execution of unconstrained rotations. Moreover, to avoid overlaps and reduce the distance between Tangram pieces, we use some traditional geometrical techniques that are applied in the automatic solution of C&P problems, which are combined with mathematical morphology operators and representations to adapt its application to the discrete domain.

During our experiments, the reduction in the complexity related to the number of piece configurations caused a decrease in the percentage of solved Tangram puzzles, which indicates that the execution of unconstrained rotations and the reflection transformation are important aspects to consider during the automatic solution of Tangram puzzles. Our method was able to find a feasible solution for most of the Tangram puzzles included in our dataset. The Tangram puzzles for which our method could not find a solution in the determined time limit were the ones identified with the numbers 7, 8, 21 and 29 in Figure 11. However, in general, we consider that our method could overcome the limitations identified in the previous methods regarding the solution of complex Tangram puzzles in feasible running time.

The results obtained show that the raster representation used to describe the Tangram pattern and the Tangram pieces also has proven efficient. Besides the loss of precision in the vector-to-raster conversion, we consider that the solutions presented after the application of the proposed method, in general, resemble the patterns informed to the method. Also, the distance transform approach was able to solve more Tangram puzzles than the other placement position strategies presented by Mundim et al. [2]. Therefore, besides the proposed method, the raster representation and the distance transform approach represent additional contributions for the area, since these approaches can be applied in the automatic solution of other two-dimensional optimization problems (e.g., placement problems, knapsack problems, cutting-stock, and bin packing problems).

For future works, we aim to extend our method to the automatic solution of C&P problems, since it can be easily adapted to solve general two-dimensional optimization problems. In order to test our raster-based mathematical morphology approach efficiency in solving C&P problems, we aim to apply our method in instances from the Special Interest Group on Cutting and Packing (ESICUP) website [22]. We intend to compare the results obtained in the tests of our method with other techniques presented in the literature, such as the works of Valle et al. [15], Dalalah et al. [16], Fischetti & Luzzi [17], Alvarez-Valdes et al. [18], and Gomes & Oliveira [19].

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] S. J. Spencer, The Tangram Route to Infinity. Trafford, 2004.
[2] L. R. Mundim, M. Andretta, M. A. Carravilla, and J. F. Oliveira, "A general heuristic for two-dimensional nesting problems with limited-size containers," International Journal of Production Research, pp. 1–24, 2017.
[3] T. Martins and M. S. G. Tsuzuki, "Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions," Expert Systems with Applications, vol. 37, no. 3, pp. 1955–1972, 2010.
[4] F. M. Yamada and H. C. Batagelo, "A comparative study on computational methods to solve tangram puzzles," Workshop of Works in Progress (WIP) in the 30th Conference on Graphics, Patterns and Images (SIBGRAPI'17), October 2017.
[5] E. S. Deutsch and K. C. Hayes Jr., "A heuristic solution to the tangram puzzle," Machine Intelligence, vol. 7, pp. 205–240, 1972.
[6] K. Oflazer, "Solving tangram puzzles: A connectionist approach," International journal of intelligent systems, vol. 8, no. 5, pp. 603–616, 1993.
[7] F. A. Andaló, G. Taubin, and S. Goldenstein, "Solving image puzzles with a simple quadratic programming formulation," in Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on. IEEE, 2012, pp. 63–70.
[8] E. D. Demaine and M. L. Demaine, "Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity," Graphs and Combinatorics, vol. 23, no. 1, pp. 195–208, 2007.
[9] D. Bartoněk, "A genetic algorithm how to solve a puzzle and its using in cartography," Acta Scientiarum Polonorum. Geodesia et Descriptio Terrarum, vol. 4, no. 2, pp. 15–23, 2005.
[10] S. Z. Kovalsky, D. Glasner, and R. Basri, "A global approach for solving edge-matching puzzles," SIAM Journal on Imaging Sciences, vol. 8, no. 2, pp. 916–938, 2015.
[11] F. M. Toledo, M. A. Carravilla, C. Ribeiro, J. F. Oliveira, and A. M. Gomes, "The dotted-board model: a new mip model for nesting irregular shapes," International Journal of Production Economics, vol. 145, no. 2, pp. 478–487, 2013.
[12] M. O. Rodrigues and F. M. Toledo, "A clique covering mip model for the irregular strip packing problem," Computers & Operations Research, vol. 87, pp. 221–234, 2017.
[13] S. MirHassani and A. J. Bashirzadeh, "A grasp meta-heuristic for two-dimensional irregular cutting stock problem," The International Journal of Advanced Manufacturing Technology, vol. 81, no. 1-4, pp. 455–464, 2015.
[14] L. R. Mundim, M. Andretta, and T. A. de Queiroz, "A biased random key genetic algorithm for open dimension nesting problems using no-fit raster," Expert Systems with Applications, vol. 81, pp. 358–371, 2017.
[15] A. M. Del Valle, T. A. De Queiroz, F. K. Miyazawa, and E. C. Xavier, "Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape," Expert Systems with Applications, vol. 39, no. 16, pp. 12 589–12 598, 2012.
[16] D. Dalalah, S. Khrais, and K. Bataineh, "Waste minimization in irregular stock cutting," Journal of Manufacturing Systems, vol. 33, no. 1, pp. 27–40, 2014.
[17] M. Fischetti and I. Luzzi, "Mixed-integer programming models for nesting problems," Journal of Heuristics, vol. 15, no. 3, pp. 201–226, 2009.
[18] R. Alvarez-Valdes, A. Martinez, and J. Tamarit, "A branch & bound algorithm for cutting and packing irregularly shaped pieces," International Journal of Production Economics, vol. 145, no. 2, pp. 463–477, 2013.
[19] A. M. Gomes and J. F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming," European Journal of Operational Research, vol. 171, no. 3, pp. 811–829, 2006.
[20] K. A. Dowsland, S. Vaid, and W. B. Dowsland, "An algorithm for polygon placement using a bottom-left strategy," European Journal of Operational Research, vol. 141, no. 2, pp. 371–381, 2002.
[21] B. Baran, B. Dogusoy, and K. Cagiltay, "How do adults solve digital tangram problems? analyzing cognitive strategies through eye tracking approach," in Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments. Springer, 2007, pp. 555–563.
[22] E. ESICUP. (2007) Special interest group in cutting and packing. Accessed on June 20th, 2019. [Online]. Available: https://www.euro-online.org/websites/esicup/data-sets/