

# Nash Meets Wertheimer: Using Good Continuation in Jigsaw Puzzles

Marina Khoroshiltseva<sup>\*,1</sup> , Luca Palmieri<sup>\*,1</sup> , Sinem Aslan<sup>1,2</sup> , Sebastiano Vascon<sup>1,2</sup> , and Marcello Pelillo<sup>1</sup> 

<sup>1</sup> Department of Computer Science (DAIS), Ca' Foscari University of Venice, Italy  
<sup>2</sup> European Center for Living Technology (ECLT), Venice, Italy

**Abstract.** Jigsaw puzzle solving is a challenging task for computer vision since it requires high-level spatial and semantic reasoning. To solve the problem, existing approaches invariably use color and/or shape information but in many real-world scenarios, such as in archaeological fresco reconstruction, this kind of clues is often unreliable due to severe physical and pictorial deterioration of the individual fragments. This makes state-of-the-art approaches entirely unusable in practice. On the other hand, in such cases, simple geometrical patterns such as lines or curves offer a powerful yet unexplored clue. In an attempt to fill in this gap, in this paper we introduce a new challenging version of the puzzle solving problem in which one deliberately ignores conventional color and shape features and relies solely on the presence of linear geometrical patterns. The reconstruction process is then only driven by one of the most fundamental principles of Gestalt perceptual organization, namely Wertheimer's *law of good continuation*. In order to tackle this problem, we formulate the puzzle solving problem as the problem of finding a Nash equilibrium of a (noncooperative) multiplayer game and use classical multi-population replicator dynamics to solve it. The proposed approach is general and allows us to deal with pieces of arbitrary shape, size and orientation. We evaluate our approach on both synthetic and real-world data and compare it with state-of-the-art algorithms. The results show the intrinsic complexity of our purely line-based puzzle problem as well as the relative effectiveness of our game-theoretic formulation.

## 1 Introduction

The problem of automatically solving jigsaw puzzles has attracted the interest of computer scientists since the early 1960's [9] and despite its intrinsic difficulty (the problem is known to be NP-hard [7]) it still remains a subject of intense studies within the computer vision and pattern recognition communities [10, 12, 32]. This is because of its applications in such diverse domains as archaeology [3, 8], genome biology [19], forensics [6], speech recognition [38], etc.

\* Equal contribution.

E-mails: [m.khoroshiltseva@unive.it](mailto:m.khoroshiltseva@unive.it), [luca.palmieri@unive.it](mailto:luca.palmieri@unive.it),  
[sinem.aslan@unive.it](mailto:sinem.aslan@unive.it), [sebastiano.vascon@unive.it](mailto:sebastiano.vascon@unive.it), [pelillo@unive.it](mailto:pelillo@unive.it)



**Fig. 1:** Three groups of fragments from an ancient broken fresco. Here, the good continuation principle plays a major role in the reassembly process.

Notwithstanding the tremendous progress made in the past decades, there still remains a large gap between the capabilities of lab-developed algorithms and the difficulties posed by real-world applications. One of the main reasons behind this state of affairs is that almost invariably existing algorithms rely only on color and/or shape information. While these clues work well on “toy” or commercial jigsaw puzzles, they often play a marginal role in real-world scenarios where individual pieces are severely deteriorated. To illustrate this point, Fig. 1 shows a few groups of fragments from an ancient shattered fresco. It is clear that here shape- and color-based compatibility functions are of little or no use in solving the puzzles. Nevertheless, we (humans) are able to effortlessly determine the relative locations of the fragments by simply “matching” the lines and curves depicted on them. The underlying principle behind this matching process is none other than Gestalt’s law of good continuation originally proposed by Wertheimer [37] which, besides being one of the most fundamental laws of visual perception [18, 28, 35], has also served as a motivation for countless computer vision algorithms [2, 24, 30].

The situation described above is in fact quite common in the wild and arises also in contexts outside the archaeological domain every time one wants to automatically reassemble shattered physical artifacts (think, e.g., of the problem of reconstructing shredded documents containing engineering or architectural drawings). We maintain that, in order to bridge the gap between the lab and the real world, puzzle-solving algorithms should make a more extensive use of this kind of clues.

The work described in this paper is a first step towards this direction. To explore the potential of purely line-based compatibility functions we introduce a new (abstract) version of the jigsaw puzzle problem in which we deliberately ignore information coming from the physical shape of the fragments as well as their color and rely instead exclusively on the presence of linear geometrical patterns. In so doing, we aim to explore the fundamental limitations of puzzle-solving algorithms in the absence of reliable color and shape information. Fig. 2 shows a simple instance of our *line-based jigsaw puzzle problem*. As can be seen, despite its small size, the problem is far from simple and, not surprisingly, existing algorithms are (by construction) totally inadequate to solve it.



**Fig. 2:** The difficulty of solving purely line-based jigsaw puzzles. The reader is invited to solve this small ( $3 \times 3$ ) instance of the problem. Our algorithm solves it perfectly.

In order to attack the problem, we develop a novel game-theoretic approach which allows one to deal with pieces of arbitrary shape, size and orientation. We formulate the general jigsaw puzzle problem as a noncooperative multiplayer game where the players are the fragments, the (pure) strategies are the planar positions and orientations, and the payoffs reflect line-based compatibilities of fragments obtained using good continuation. Under this framework it is easy to see that puzzle solutions correspond to Nash equilibria of this “jigsaw puzzle game” [21, 22] and in order to find them we use classical multi-population replicator dynamics from evolutionary game theory [13, 36].

We evaluate our approach on both synthetic and real-world datasets constructed specifically for this work and compare it with state-of-the-art algorithms. The experimental results show the intrinsic complexity of our purely line-based puzzle problem, confirm the inadequacy of existing approaches, and demonstrate the relative effectiveness of our game-theoretic formulation.

## 2 Related Works

The literature on puzzle-solving has predominantly focused on square jigsaw puzzles, where only pictorial information is available due to the identical shapes of all pieces, e.g., [1, 5, 10, 17, 25, 27, 32, 33]. Earlier studies such as [5, 27] tackled square jigsaw puzzles with known piece size and orientation. Cho et al. [5] used loopy belief propagation for graph-based piece arrangement, requiring low-resolution image estimations. Pomeranz et al. [27] introduced a prediction-based dissimilarity metric and best-buddy concept for fully automatic solutions. Gallagher et al. [10] addressed puzzles with unknown piece orientation, introducing a tree-based algorithm and a compatibility metric based on Mahalanobis distance. Paikin and Tal [23] improved accuracy by modifying the dissimilarity metric of Pomeranz et al. [27] and strategically selecting an anchor piece. Son et al. [32] dealt with noise in pictorial content, and unknown piece orientation and puzzle size by requiring compatibility consensus within puzzle piece loops, extending beyond immediate neighbors. Besides these, a few works addressed solving puzzles with irregularly shaped fragments [8, 12], which hold potential applications in real-world scenarios such as archaeological puzzle solving. Derech and Tal [8] addressed archaeological puzzles in cultural heritage, characterized by irregularly shaped pieces prone to color fading, abrasion, and discontinuity, along with degraded fragment boundaries. They suggested pre-extrapolating each fragment and introduced confidence in dissimilarity scores, considering the uniqueness of

match and fragment size during placement. Harel and Ben-Shahar [12] introduced a new synthetically created puzzle type, namely the Crossing Cuts puzzle, and abstracted the problem as a multi-body spring-mass dynamical system supported with hierarchical loop constraints.

While certain works, e.g., [8, 12, 23, 27, 32], have reported effective performances, they all employed greedy approaches, making decisions based only on local matchings without considering global consequences. Andalo et al. [1] was the first to formulate the problem as a global optimization problem, utilizing a Constrained Gradient Ascent algorithm for its solution. Khoroshiltseva et al. [17] adopted a nonlinear relaxation labeling (RL) approach, extending the labeling update rule by Rosenfeld et al. [29], encouraging but without guaranteeing feasible solutions. Vardi et al. [34] proposed a multi-phase approach for relaxation labeling, ensuring convergence to feasible puzzle solutions, even for puzzles with both known and unknown piece orientation. All these works adopting global optimization [1, 17, 34] were designed specifically for the squared jigsaw puzzles with known puzzle sizes; with the assumption of known piece orientation in [1, 17].

In the last years, there has been an increased interest in neural network methods for puzzle solving. A common issue with learning-based approaches is that many struggle with scalability, performing well only on small-scale puzzles. For instance, Paumard et al. [25] proposed a CNN model to predict neighboring relationships in 9-piece puzzles. Song et al. [33] introduced the Siamese-Discriminant Deep Reinforcement Learning method for 9-piece puzzles with eroded boundaries. More recently, diffusion models have begun to be utilized in puzzle-solving. These models involve introducing time-dependent noise into each piece’s translation and rotation during the forward process, followed by iterative denoising during the backward process to reconstruct the original image. Hosseini et al. [14] introduced PuzzleFusion for puzzles with up to 16 irregular-shaped fragments. Scarpellini et al. [31] represented puzzles as graphs and processed them with an Attention-based Graph Neural Network within a diffusion model framework.

The choice of compatibility measurement significantly impacts puzzle-solving performance. While square jigsaw puzzles typically compare pictorial information along piece boundaries, irregularly shaped puzzles also consider matching fragment shapes. Various approaches exist: some compute color differences along fragment boundaries [1, 5, 23, 27], while others consider distributions of local gradients [10, 32]. Some works compare the length of matching boundaries between fragments to assess their shapes [12], while others combine all these measurements [8]. This study is the first, to our knowledge, to specifically address purely line-based compatibilities.

### 3 Jigsaw Puzzles and Nash Equilibria

#### 3.1 Nash Equilibria

We start this section by introducing some fundamental concepts from game theory. More details can be found in standard textbooks such as [22, 36].

Formally, a (noncooperative) game involves a set of *players*  $\mathcal{P} = \{1, \dots, n\}$ , each associated with a set of *pure strategies*  $S_i = \{1, \dots, m_i\}$  ( $i = 1 \dots n$ ). A vector of pure strategies  $s = (s_1, \dots, s_n)^T$  is called a *strategy profile* and the set  $S = S_1 \times S_2 \times \dots \times S_n$  is the *pure strategy space* of the game. The (*combined*) *payoff function* is a function  $\pi : S \rightarrow \mathbb{R}^n$  which assigns a real valued payoff  $\pi_i(s) \in \mathbb{R}$  to each player  $i \in \mathcal{P}$  and (pure) strategy profile  $s \in S$ . This is the payoff that player  $i$  gets when players play (pure) strategies  $s_1, \dots, s_n$ .

A *mixed strategy* for player  $i \in \mathcal{P}$  is simply a probability distribution over its pure strategy set  $S_i$ . Geometrically, it is a vector  $x_i = (x_{i1}, \dots, x_{im_i})^T$  belonging to the *standard simplex*:

$$\Delta_i = \left\{ x_i \in \mathbb{R}^{m_i} : \sum_{h=1}^{m_i} x_{ih} = 1, \text{ and } x_{ih} \geq 0 \text{ for all } h \in S_i \right\} .$$

Each component  $x_{ih}$  denotes the probability with which player  $i$  will choose its pure strategy  $h$ . Accordingly, a *mixed strategy profile*  $x = (x_1, \dots, x_n)^T$  is defined as a vector of mixed strategies, each  $x_i \in \Delta_i$  representing the mixed strategy assigned to player  $i \in \mathcal{P}$ . Each mixed strategy profile lives in the *mixed strategy space* defined as  $\Theta = \Delta_1 \times \Delta_2 \times \dots \times \Delta_n$ .

In non-cooperative game theory we make the assumption that players choose their strategies in total ignorance of the other players' choices. Under this independence assumption, the probability that a given pure strategy profile  $s \in S$  will be used when mixed strategy profile  $x \in \Theta$  is played becomes:  $x(s) = \prod_{i=1}^n x_{is_i}$  and hence the *expected payoff* for player  $i$  associated to  $x \in \Theta$  can be written as:

$$u_i(x) = \sum_{s \in S} x(s)\pi_i(s). \tag{1}$$

Following standard game-theoretic notations, let  $(x_i, y_{-i}) \in \Theta$  denote the strategy profile where player  $i$  plays strategy  $x_i \in \Delta_i$  whereas the other players  $j \in \mathcal{P} \setminus \{i\}$  play based on the strategy profile  $y \in \Theta$ . A mixed strategy profile  $x^* = (x_1^*, \dots, x_n^*) \in \Theta$  is said to be a *Nash equilibrium* if no player has an incentive to unilaterally deviate from it or, more formally, if

$$u_i(x_i^*, x_{-i}^*) \geq u_i(x_i, x_{-i}^*) \tag{2}$$

for all  $i \in \mathcal{P}$  and  $x_i \in S_i$ . If the inequalities hold strictly,  $x^*$  is called a *strict* Nash equilibrium. Nash equilibrium is a key concept in game theory. In his celebrated paper [21], John Nash proved that any non-cooperative game has at least one (mixed) Nash equilibrium.

### 3.2 Jigsaw Puzzle Games

We now cast the puzzle-solving problem as a non-cooperative multiplayer game.

According to our formulation, the set of players  $\mathcal{P}$  coincides with the set of all puzzle pieces, and the set of pure strategies associated to each player is the set of all possible triplets  $s_i = (x_i, y_i, \theta_i)$ , where  $(x_i, y_i)$  refers to planar coordinates

and  $\theta_i$  to a rotation angle. Of course, to reduce the computational burden, in our experiments we discretize the strategy space and make a number of simplifying assumptions (see next section for details).

In this paper, we suppose that the payoffs associated to each player are “additively separable,” i.e., that for a pure strategy profile  $s = (s_1, \dots, s_n) \in S$ , the payoff of every player  $i \in \mathcal{P}$  can be written as:

$$\pi_i(s) = \sum_{j=1}^n A_{ij}(s_i, s_j) \quad (3)$$

where  $A_{ij}$  is the *partial payoff* matrix between players  $i$  and  $j$ . This makes the proposed game a member of a well-studied subclass of multi-player games, known as *polymatrix games* [16].

Here, the payoff function reflects the consensus between pieces given their chosen positions and orientation. Accordingly, given two pieces  $i, j \in \mathcal{P}$  and two pure strategies  $s_i = (x_i, y_i, \theta_i)$  and  $s_j = (x_j, y_j, \theta_j)$ , the payoff  $A_{ij}(s_i, s_j)$  measures the compatibility of the following two hypotheses:  $H_1$ : “piece  $i$ , rotated by  $\theta_i$  degrees, is in position  $(x_i, y_i)$ ” and “piece  $j$ , rotated by  $\theta_j$  degrees, is in position  $(x_j, y_j)$ ”.

Once the jigsaw-puzzle problem is formulated in this way, it is easy to see that Nash equilibria correspond to plausible puzzle solutions. This is indeed motivated by a formal connection established by Miller and Zucker [20] between Nash equilibria of polymatrix games and the notion of a consistent labeling in relaxation labeling processes [15, 26] (see also [17]).

### 3.3 Finding Nash Equilibria

To find a Nash equilibrium of our jigsaw puzzle game we use a well-known class of game dynamics studied in evolutionary game theory [13, 36]. Under this framework one imagines that the game is played repeatedly, generation after generation, in a population of players and that a selection process acts on the strategy set, resulting in the evolution of the fittest strategies.

This selection dynamics is commonly modeled by (continuous-time) multi-population *replicator dynamics* inspired by Darwin’s principle of natural selection. In our polymatrix game setting, this can be written as:

$$\dot{x}_{ih} = x_{ih} \left( \pi_{ih}(x) - \sum_k x_{ik} \pi_{ik}(x) \right) \quad (4)$$

where  $i \in \mathcal{P}$  and  $h \in S_i$  is a pure strategy associated to player  $i$ . Here,

$$\pi_{ih}(x) = \sum_j \sum_k x_{jk} A_{ij}(h, k) \quad (5)$$

measures the expected payoff of player  $i$  when playing strategy  $h$  and  $\sum_k x_{ik} \pi_{ik}(x)$  is the average population payoff.

The following theorem states that interior limits points of (4) are in one-to-one correspondence to Nash equilibria (see, e.g., [36] for a proof). This motivates the use of replicator dynamics to find the Nash equilibria of our puzzle games.

**Theorem 1.** *A point  $x \in \Theta$  is a limit trajectory of (4) starting from the interior of  $\Theta$  if and only if  $x$  is a Nash equilibrium. Further, if point  $x \in \Theta$  is a strict Nash equilibrium then it is also asymptotically stable.*

In our experiments, we used the following discrete-time counterpart of (4),

$$x_{ih}(t + 1) = x_{ih}(t) \frac{\pi_{ih}(x(t))}{\sum_k x_{ik}(t) \pi_{ik}(x(t))} \tag{6}$$

which has the same dynamical properties as the continuous one (see [26] for a detailed analysis).

The computational complexity of finding a Nash equilibrium of a our puzzle game using (6) is  $\mathcal{O}(kcn^2)$ , where  $n$  is the number of players (pieces),  $c$  is the number of pure strategies (position/orientation) and  $k$  is the number of iterations. In the following section we shall see how to dramatically reduce the computational complexity of the algorithm by using a proper discretization and sparsification strategy.

## 4 Payoff Functions

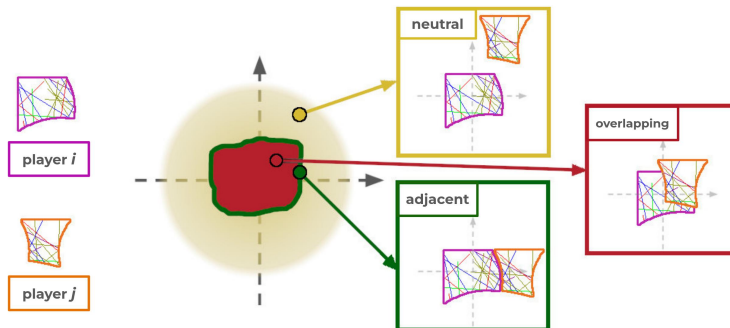
### 4.1 Partial Payoff Matrix

To reduce the strategic space, we define for each pair of players a proximity space  $\Gamma$ . In this space the player  $i$  assumes the strategy  $s_0 = (0, 0, 0)$ , while the player  $j$  translates and rotates around  $i$ . The space is divided into three regions: *adjacent*, *overlapping* and *neutral*. The region where players  $i$  and  $j$  maintain contact without overlapping denotes “legal” positioning, positively contributing to players’ gain. On the contrary, the area in which the players overlap is considered an “illegal” positioning and corresponds to a negative contribution - penalizing the player’s payoff. Other positions, where players are not in direct contact, are labeled “neutral” and do not explicitly affect players’ payoffs. The concept is illustrated in Fig. 3.

We therefore define the reduced *partial payoff* matrix  $\tilde{A}_{ij}$  which contains the payoffs related to the relative strategies  $\gamma \in \Gamma$  of player  $j$  against the fixed strategy of player  $i$ . The payoff  $\tilde{A}_{ij}(\gamma)$  is formally expressed in Eq. 7:

$$\tilde{A}_{ij}(\gamma) = \begin{cases} R_{ij\gamma} & (i \neq j) \wedge (\gamma \in \textit{adjacent region}) \\ -1 & (i = j) \vee (\gamma \in \textit{overlapping region}) \\ 0 & (i \neq j) \wedge (\gamma \in \textit{neutral region}) \end{cases} \tag{7}$$

The compatibility score  $R_{ij\gamma}$  evaluates the placement when player  $j$  chooses the strategy  $\gamma$  and can be calculated based on various factors such as appearance (e.g., colors or line continuity), geometry, or semantic proximity. Here we focus on the line-based compatibility.



**Fig. 3:** The three regions of the space: neutral (yellow), adjacent (green) and overlapping (red).

## 4.2 Line-based Compatibility Measure

The compatibility score  $R_{ij\gamma}$  reflects the quality of the match based on the characteristics of the two players, approaching 1 for the best strategy and approaching 0 for strategies that do not significantly benefit the players. We decide to use the lines as characteristics of each player in the game and calculate the compatibility score  $R_{ij\gamma}$  based on lines only.

The score is calculated according to the Gestalt principle of good continuation [18] along the lines of both players. The compatibility score between two players  $i$  and  $j$  for a relative strategy  $\gamma$  is high if the lines of  $j$  are continuations of the lines of  $i$  and vice versa. A line  $r$  of a player  $i$  denoted as  $l_{i,r}$  is represented by a start and end point, an angle  $\alpha_r$  and a semantic category  $c_r$ .

First, we estimate the cost as a measure of the effort needed to continue a pair of lines, which we denote as  $C_L$  and describe in Eq. 8. This cost is zero when the lines perfectly continue each other and reaches high values when the lines are not aligned.

$$C_L(l_{i,r}, l_{j,s}) = \begin{cases} \text{dist}(l_{i,r}, l_{j,s}) & \text{if } |\alpha_r - \alpha_s| < \epsilon_a \wedge c_r = c_s \\ \text{cost}_{max} & \text{otherwise} \end{cases} \quad (8)$$

The  $\text{dist}(l_{i,r}, l_{j,s})$  is the smallest distance between any of the two end points of each line,  $\text{cost}_{max}$  is a constant value related to the puzzle,  $\epsilon_a$  is the tolerance of the angle between two lines.

The total cost of matching the set of lines of player  $i$  (denoted as  $L_i$ ) and those of player  $j$  (denoted as  $L_j$ ) is estimated as described in Eq. 9. The first component  $C_{\text{LAP}}$  is obtained by solving the associated Linear Assignment Problem (LAP) [4] and the second  $\phi_{ij}$  is a penalty term for the unmatched lines.

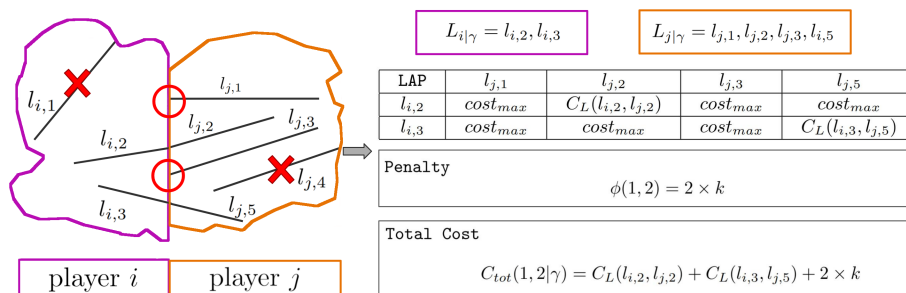
$$C_{tot}(L_i, L_j|\gamma) = C_{\text{LAP}}(L_i, L_j|\gamma) + \phi_{ij} \quad (9)$$



The cost of the solution **LAP** consists of the sum of the costs of matching pairs of lines selected in such a way that the cost of the assignment is minimized.

$$C_{\mathbf{LAP}}(L_i, L_j | \gamma) = \sum_{r \in (1, |L_i|)} C_L(l_{i,r}, l_{j,s(r)} | \gamma) \quad (10)$$

We denote the assignment solution as  $s(r)$ , which associates the  $s(r)$ -line in  $L_j$  to the  $r$ -line in  $L_i$ . We use the  $C(\cdot | \gamma)$  notation to dynamically create the sets of lines such as  $L_{i|\gamma}$  by selecting only the subset of lines of  $L_i$  which are coherent to the current relative strategy  $\gamma$  (lines  $l_{i,1}$  and  $l_{j,4}$  of example in Fig. 4). The **LAP** solution assigns a number of matches based on the cardinality of the smallest set. In the presence of unassigned lines we penalize the solution by adding the penalty term  $\phi = u_l \cdot k$ , where  $k$  is the constant factor and  $u_l$  is the number of unassigned lines.



**Fig. 4:** An example of the computation of the cost for the players  $i$  and  $j$  given a candidate  $\gamma$  strategy. The two sets  $L_{i|\gamma}$  and  $L_{j|\gamma}$  are created by selecting only lines which intersects the opposite player, and the total cost consists of the sum of the cost of the corresponding **LAP** and the penalty for unmatched lines.

The compatibility score is a measure of the quality of the continuation of all lines of both players given the relative strategy  $\gamma$ . Therefore, it will be inversely proportional to the total cost of matching the two sets, as described in Eq. 11:

$$R_{ij\gamma} = 1 - \frac{\min(C_{tot}(i, j, \gamma), \tau)}{\tau} \quad (11)$$

The cost is truncated and normalized using  $\tau$  which is based on the best compatibility scores obtained by all players. In doing so, we discard the strategies that obtained lower scores, sparsifying the partial payoff matrix  $\tilde{A}_{ij}(\gamma)$  and reducing the computational complexity.

## 5 Experiments

To demonstrate the effectiveness and flexibility of our method, we conducted an extensive evaluation on different types of puzzles. Concretely, we evaluate our

approach on: (1) puzzles with *squared pieces with known piece orientation*, (2) puzzles with *polygonal shapes with unknown piece orientation*, (3) puzzles with *irregular shapes with unknown piece orientation*.

For all categories, we experimented with synthetic data generated by randomly drawing lines on blank images and real data in the form of vector-based geographical maps.

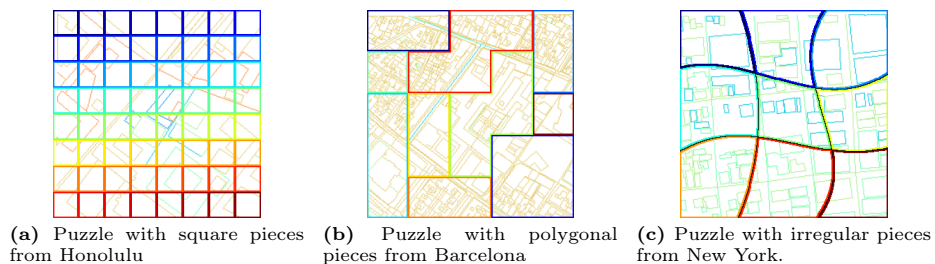
## 5.1 Datasets

The creation of a large scale dataset for puzzle solving is a challenging task and there is not a standard dataset with different types of pieces for the puzzle solving community. We decided to use two sets of puzzles to evaluate our approach: synthetically generated puzzles and puzzles obtained cutting real-world data into pieces.

**Synthetic Lines Generation** We developed a puzzle generator pipeline to create a dataset with fine-grained control on different scenarios, including the number of pieces, their shapes (squared, polygonal, irregular), and the characteristics of lines (number and types of lines) in an image.

For the puzzles used in this work, we created images with 50 random lines, each assigned a semantic category. In the baseline scenario of squared pieces, we created 30 images using a single semantic category. In the polygonal and irregular scenarios, we created 10 images using 5 different categories rendered with distinct colors on the associated images.

**Puzzles from Real Data** Since we tackle the novel task of *line-based* puzzle solving, we created a dataset that consists of vector maps. Vector maps are a real-world example of line-based real-world data that exhibit a wide variety of line types and shapes. We gathered vector maps from Openstreetmap<sup>3</sup> to create a test set of 10 images. We processed them to extract coordinates, angles and the semantic category (related to the mapped object) and converted them to raster images to create the puzzle images.



**Fig. 5:** Some examples of puzzles using real vector maps<sup>3</sup>.

<sup>3</sup> Map data from OpenStreetMap: <https://www.openstreetmap.org>

**Pieces Creation** The puzzles aim to be a benchmark for the evaluation of different types of algorithms, therefore we created 3 types of puzzle pieces. We created squared pieces by evenly dividing the image space, polygonal pieces using handcrafted patterns (for a finer control on the solution) and irregular pieces by partitioning the image using horizontal and vertical curves. Example of puzzles created from real data using these techniques are visible in Fig. 5.

## 5.2 Evaluation Metrics

In the squared pieces environment, we use known metrics [5, 27], namely the direct (**D**) metric, which compute the proportion of correctly positioned pieces and the neighbors (**N**) metric, which compute the ratio of correct relative neighbors. For the irregular puzzles, we report only the direct (**D**) metric modified to accept the correct assembly with a tolerance threshold.

## 5.3 Results

The performance obtained in the different datasets demonstrates the generalization capability and the accuracy of our method in the puzzle solving task.

Algorithm	Synthetic		Maps	
	D (↑)	N (↑)	D (↑)	N (↑)
Ours	<b>0.99</b>	<b>0.99</b>	<b>0.203</b>	<b>0.268</b>
Pomeranz et al. [27]	0.014	0.241	0.020	0.158
Paikin et al. [23]	0.022	0.197	0.060	0.208
Positional Diffusion <sup>4</sup>	0.072	-	0.035	-

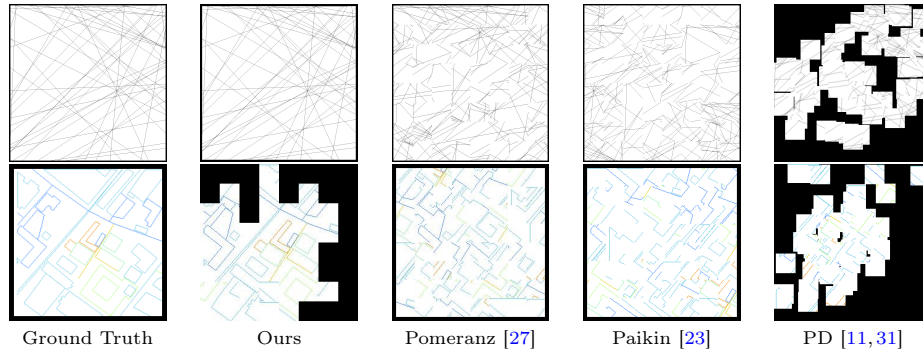
**Table 1:** Quantitative results on puzzles with squared pieces.

We compare our approach with Pomeranz et al. [27], Paikin et al. [23] and Positional Diffusion<sup>4</sup> [11] for squared pieces with known orientation in Table 1 and with Derech et al. [8] for arbitrarily shaped pieces (polygonal and irregular) with unknown orientation in Table 2. To ensure a fair comparison, we need to take into consideration some of the differences between the algorithms, since previous works were designed to tackle colored images. As a general rule, we render our lines with different colors based on the semantic category to enable a possible color match. Moreover, in the squared pieces scenario, we feed the input image (therefore fixing the final shape of the puzzle) to both competitor approaches, while in our case we just fix one anchor and solve the puzzle around it. In the irregular scenario, the method proposed by Derech et al. requires an

<sup>4</sup> We use their implementation (available at the time) for the results, which does not provide values for the Neighbors metric.

extrapolated version of each piece: we built a *perfect* extrapolated version for all pieces in all puzzles to allow a correct match between overlapping parts. Even with these settings, our approach is the only one able to achieve a satisfactory solution on the different puzzles.

**Squared Pieces** The performance obtained for puzzles with square pieces are presented in Table 1. Our approach significantly outperforms competitors in terms of all metrics, establishing a new baseline for this novel task.



**Fig. 6:** Qualitative results on squared pieces on both synthetic and real data. Our reconstructed puzzles are cropped for a better visualization.

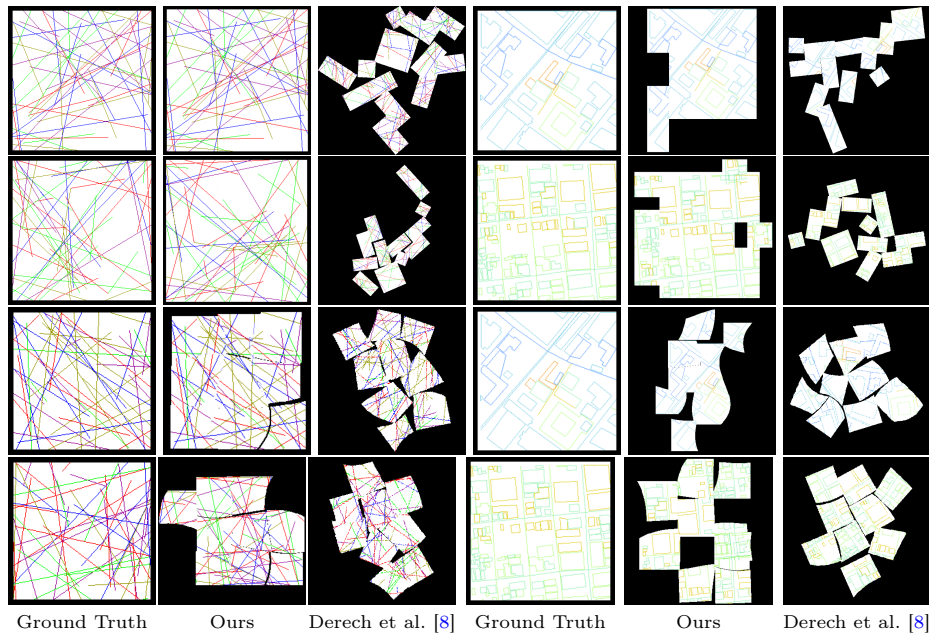
The puzzles from vector maps images were created as a novel challenging benchmark, alternating textureless regions with small area with a large density of lines and creating a challenge for any kind of reassembly algorithm. The accuracy of our approach drops significantly in these puzzles, primarily due to the effect of early wrong choices during reconstruction.

Algorithm	Polygonal Pieces		Irregular Pieces	
	Synthetic Maps		Synthetic Maps	
	D (↑)	D (↑)	D (↑)	D (↑)
Ours (90° rotations)	<b>0.99</b>	<b>0.51</b>	<b>0.622</b>	<b>0.283</b>
Derech et al. [8]	0.25	0.21	0.11	0.11

**Table 2:** Quantitative results on polygonal and irregular pieces.

**Polygonal and Irregular Pieces** Our approach tackles without modification any kind of puzzle, and we report the performance on irregular pieces in Table 2. The accuracy hints that our method is a step in the right direction while the approach from Derech et al. [8] struggle to reconstruct the puzzle in both the polygonal and the irregular case.

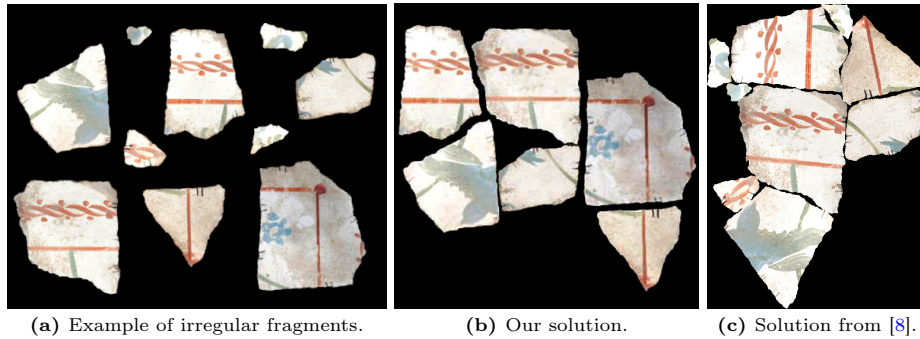
Our accuracy is significantly higher in the polygonal case, due to the exact definition of the grid. Due to the handcrafted nature of the polygonal pieces, we can guarantee that the correct transformation lies within the grid used to define the strategies of our players. This is exploited by our algorithm which obtains an almost perfect score on the synthetic polygonal case and correctly assembles more than half of the polygonal pieces, scoring even better in the case of squared pieces on the same maps. Although the polygonal pieces offer shape clues, the method by Derech et al. [8] fails to assemble them.



**Fig. 7:** Qualitative results on polygonal (first two rows) and irregular pieces (last two rows). Reconstructed puzzles are cropped for a better visualization.

In the irregular case, our accuracy is lower, pointing out that the players did not choose the best strategy, or the pieces were not in the absolute correct position. Although our method achieves sub-optimal accuracy in this scenario, we see that the alternative solutions found are plausible. Visual inspection of the reconstructions gives us further insights: our approach finds plausible solutions *which satisfy the principle of good continuation of the lines* without solving the puzzle in its original squared form, resulting in *wrong alignment with respect to the metric*. We want to strengthen here that our algorithm does not know the final shape of the puzzle, as it was designed for a real-world scenario, where the shape of the final puzzle is unknown.

Finally, we provide our results on a case study consisting of a puzzle composed of real fragments that archaeologists have been studying for many years in Figure



**Fig. 8:** An example of a real archaeological puzzle composed of irregular fragments.

**8a.** On this small puzzle, archaeologists annotated the patterns which we used as features for our players in a game without rotation. It is seen that the law of good continuation seems to be the wisest choice, as shown by the solution that our approach obtains, in Figure 8b. This result confirms that our algorithm is a first step towards the solution of complex real-world puzzles, whereas the method by Derech et al. [8] for archaeological puzzles struggles due to the low discrimination power of color and shape in this case study.

**Limitations** The challenging datasets highlight some limitations of our algorithm. The discretization of the space used to create the partial payoff relative transformation is a strong bottleneck when scaling to larger puzzles, making it a computationally expensive process. Moreover, the variable density of the lines and the presence of empty regions mislead our algorithm in some of the map puzzles, alternating success and failure cases. These limitations will be tackled in future works.

## 6 Conclusions

Motivated by the observation that in real-world applications color and shape features used by existing jigsaw puzzle-solving algorithms are often unusable, in this paper we proposed a new challenging version of the problem which relies solely on the presence of linear geometrical patterns. To address this problem, we developed a principled game-theoretic framework which allows one to solve puzzles with pieces of arbitrary shape, size and orientation. Experimental results show the inadequacy of standard techniques as well as the relative advantages of our game-theoretic formulation. Future work will focus on extending the proposed ideas to more complex curve clues and on integrating good continuation features with standard shape and color information.

**Acknowledgements** This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 964854.

## References

1. Andalo, F.A., Taubin, G., Goldenstein, S.: PSQP: Puzzle solving by quadratic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(2), 385–396 (2016)
2. Ben-Shahar, O., Zucker, S.: General geometric good continuation: From Taylor to Laplace via level sets. *Int. J. Comput. Vision* **86**(1), 48–71 (2010)
3. Brown, B.J., Toler-Franklin, C., Nehab, D., Burns, M., Dobkin, D., Vlachopoulos, A., Dumas, C., Rusinkiewicz, S., Weyrich, T.: A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings. *ACM Transactions on Graphics* **27**(3), 1–9 (2008)
4. Brualdi, R.A.: *Combinatorial Matrix Classes*. Cambridge University Press (2006)
5. Cho, T.S., Avidan, S., Freeman, W.T.: A probabilistic image jigsaw puzzle solver. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 183–190 (2010)
6. Deever, A., Gallagher, A.: Semi-automatic assembly of real cross-cut shredded documents. In: *IEEE international Conference on Image Processing*. pp. 233–236 (2012)
7. Demaine, E.D., Demaine, M.L.: Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics* **23**(Suppl 1), 195–208 (2007)
8. Derech, N., Tal, A., Shimshoni, I.: Solving archaeological puzzles. *Pattern Recognition* **119**, 108065 (2021)
9. Freeman, H., Garder, L.: Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers* **13**, 118–127 (1964)
10. Gallagher, A.C.: Jigsaw puzzles with pieces of unknown orientation. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 382–389 (2012)
11. Giuliani, F., Scarpellini, G., James, S., Wang, Y., Del Bue, A.: Positional diffusion: Ordering unordered sets with diffusion probabilistic models. *arXiv preprint arXiv:2303.11120* (2023)
12. Harel, P., Ben-Shahar, O.: Crossing cuts polygonal puzzles: Models and solvers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3084–3093 (2021)
13. Hofbauer, J., Sigmund, K.: *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, UK (1988)
14. Hossieni, S.S., Shabani, M.A., Irandoust, S., Furukawa, Y.: Puzzlefusion: Unleashing the power of diffusion models for spatial puzzle solving. *Advances in Neural Information Processing Systems* **36** (2024)
15. Hummel, R.A., Zucker, S.W.: On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **5**, 267–287 (1983)
16. Janovskaya, E.B.: Equilibrium points in polymatrix games. *Litovskii Matematicheskii Sbornik* **8**, 381–384 (1968), in Russian
17. Khoroshiltseva, M., Vardi, B., Torcinovich, A., Traviglia, A., Ben-Shahar, O., Pelillo, M.: Jigsaw puzzle solving as a consistent labeling problem. In: *Computer Analysis of Images and Patterns (CAIP)*. pp. 392–402. Springer (2021)
18. Koffka, K.: *Principles of Gestalt Psychology*. A Harbinger Book (1963)
19. Marande, W., Burger, G.: Mitochondrial DNA as a genomic jigsaw puzzle. *Science* **318**(5849), 415–415 (2007)

20. Miller, D.A., Zucker, S.W.: Copositive-plus Lemke algorithm solves polymatrix games. *Operations Research Letters* **10**(5), 285–290 (1991)
21. Nash, J.: Non-cooperative games. *Annals of mathematics* pp. 286–295 (1951)
22. Osborne, M.J., Rubinstein, A.: *A Course in Game Theory*. MIT Press, Cambridge, MA (1994)
23. Paikin, G., Tal, A.: Solving multiple square jigsaw puzzles with missing pieces. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4832–4839 (2015)
24. Pasupathy, A., Peterson, M.A., Bernhardt-Walther, D., Elder, J. (eds.): *Perceptual Organization in Computer and Biological Vision*. *Frontiers in Computer Science—Computer Vision* (2024)
25. Paumard, M.M., Picard, D., Tabia, H.: Deepzlle: Solving visual jigsaw puzzles with deep learning and shortest path optimization. *IEEE Transactions on Image Processing* **29**, 3569–3581 (2020)
26. Pelillo, M.: The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision* **7**, 309–323 (1997)
27. Pomeranz, D., Shemesh, M., Ben-Shahar, O.: A fully automated greedy square jigsaw puzzle solver. In: *CVPR 2011*. pp. 9–16. IEEE (2011)
28. Rock, I., Palmer, S.: The legacy of Gestalt psychology. *Scientific American* **263**(6), 84–91 (1990)
29. Rosenfeld, A., Hummel, R.A., Zucker, S.W.: Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics* pp. 420–433 (1976)
30. Sarkar, S., Boyer, K.L.: *Computer Perceptual Organization in Computer Vision*. World Scientific (1994)
31. Scarpellini, G., Fiorini, S., Giuliari, F., Morerio, P., Del Bue, A.: Diffassemble: A unified graph-diffusion model for 2D and 3D reassembly. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2024)
32. Son, K., Hays, J., Cooper, D.B.: Solving square jigsaw puzzle by hierarchical loop constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(9), 2222–2235 (2018)
33. Song, X., Jin, J., Yao, C., Wang, S., Ren, J., Bai, R.: Siamese-discriminant deep reinforcement learning for solving jigsaw puzzles with large eroded gaps. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 37, pp. 2303–2311 (2023)
34. Vardi, B., Torcinovich, A., Khoroshiltseva, M., Pelillo, M., Ben-Shahar, O.: Multi-phase relaxation labeling for square jigsaw puzzle solving. In: *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, (VISIGRAPP)*. pp. 785–795 (2023)
35. Wagemans, J., Elder, J.H., Kubovy, M., Palmer, S., Peterson, M.A., Singh, M., von der Heydt, R.: A century of Gestalt psychology in visual perception I. Perceptual grouping and figure-ground organization. *Psychological Bulletin* **138**(6), 1172–1217 (2012)
36. Weibull, J.W.: *Evolutionary Game Theory*. MIT press, Cambridge, MA (1997)
37. Wertheimer, M.: Untersuchungen zur lehre von der gestalt. II. *Psychologische Forschung* **4**, 301–350 (1923), translation published in W. Ellis (1938). *A source book of Gestalt psychology* (pp. 71–88). London: Routledge & Kegan Paul
38. Zhao, Y.X., Su, M.C., Chou, Z.L., Lee, J.: A puzzle solver and its application in speech descrambling. In: *WSEAS International Conference on Computer Engineering and Applications*. pp. 171–176 (2007)