Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01440-0

ORIGINAL ARTICLE

Solving puzzles using knowledge-based automation: biomimicry of human solvers

Syifa Fauzia[1] · Sean Chen[2] · Ren-Jung Hsu[1] · Rex Chen[3] · Chi-Ming Chen[1]

Received: 9 January 2024 / Accepted: 31 March 2024
© The Author(s) 2024

Abstract

The human brain's remarkable efficiency in solving puzzles through pictorial information processing serves as a valuable inspiration for computational puzzle solving. In this study, we present a nucleation algorithm for automated puzzle solving, developed based on statistical analysis of an empirical database. This algorithm effectively solves puzzles by choosing pieces with infrequent and iridescent edges as nucleation centers, followed by the identification of neighboring pieces with high resemblances from the remaining puzzle pieces. For the 8 different pictures examined in this study, both empirical data and computer simulations consistently demonstrate a power-law relationship between solving time and the number of puzzle pieces, with an exponent less than 2. We explain this relationship through the nucleation model and explore how the exponent is influenced by the color pattern of the puzzle picture. Moreover, our investigation of puzzle-solving processes reveals distinct principal pathways, akin to protein folding behavior. Our study contributes to the development of a cognitive model for human puzzle solving and color pattern recognition.

Keywords Automated puzzle solving · Knowledge-based automation · Color pattern recognition · nucleation model

Introduction

The primary scientific objective of artificial intelligence (AI) is to comprehend the underlying principles that drive intelligent behaviors in natural systems and utilize this knowledge to construct artificial systems capable of matching natural abilities. Achieving this goal involves two distinct approaches in developing automated algorithms for enabling intelligent behaviors. One approach focuses on achieving high performance in specific domains by employing methods such as brute-force, optimization-oriented, or domain-specific solutions. A prime example of this performance-driven approach is the program AlphaGo, which surpassed professional Go players by combining Monte Carlo tree search, value and policy networks, and reinforcement learning [1]. This type of AI does not aim to replicate human functioning but can excel in its designated tasks [2–7]. The other approach seeks to mimic human behavior to attain comparable performance on cognitive tasks and comprehend the underlying principles [8–11]. Some researchers have even attempted to create computer models of the cognitive architecture of the human mind, a fascinating and crucial area of AI [12]. Notable cognitive architectures include ACT-R [13], Soar [14], and PRODIGY [15].

In this study, our objective is to investigate color pattern recognition and decision-making processes employed by human puzzle solvers. Puzzle solving has long been a beloved pastime, encompassing various types of puzzles like jigsaw puzzles, edge matching puzzles, and polyomino packing puzzles. Interestingly, all three puzzle types are considered NP-complete, and they can be converted into equivalent versions of each other [16]. This has piqued the interest of researchers from different fields, including computer vision, pattern recognition, and image processing, who have been exploring the possibilities of automating puzzle solving using computers.

The automatic puzzle-solving domain holds great potential for diverse applications. For instance, it can be employed

Syifa Fauzia and Sean Chen contribute equally.

✉ Chi-Ming Chen
cchen@phy.ntnu.edu.tw

[1] Department of Physics, National Taiwan Normal University, Taipei, Taiwan

[2] Faculty of Science, University of British Columbia, Vancouver, BC, Canada

[3] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Published online: 16 May 2024

in tasks such as speech descrambling [17], image descrambling [18], assembly of cracked oil paintings [19], reassembling archeological remnants [20] and document fragments [21], DNA/RNA modeling [22], and molecular docking for drug design [23]. The study of automated puzzle solving has opened up new avenues for innovative applications and has garnered significant interest and attention across diverse scientific disciplines.

The aim of our study is to create models that depict human perception of color patterns and problem-solving skills. We have developed an automated solver for rectangular piece puzzles that mimics the problem-solving approaches used by human solvers. Our investigation revolves around two core research inquiries: (1) What factors guide human solvers when choosing initial puzzle pieces? and (2) How skilled are human solvers in identifying color patterns on puzzle pieces? By tackling these questions, we aspire to enhance our understanding of the cognitive mechanisms underlying puzzle solving and contribute to the progress of automated puzzle solvers that replicate human-like strategies.

The subsequent sections of this paper are structured as follows. Firstly, we provide an overview of the puzzle problem and review existing research on solving puzzles. Secondly, in Section "Methods", we present a nucleation model of puzzle solving, followed by a detailed statistical analysis of edge features of puzzle pieces using a puzzle-solving database of human solvers. Additionally, we introduce a computer simulation algorithm designed to replicate human puzzle-solving processes based on the nucleation model and this statistical analysis. Moving on, Section "Results and Discussion" discusses the outcomes of computer simulations, focusing on the relationship between the average puzzle solving time and the number of puzzle pieces, as well as visualizations of the general puzzle-solving process. Finally, in Section "Conclusions", we present our conclusions based on the findings and discuss the implications of this research.

Problem definition

The puzzle problem involves correctly assembling all puzzle pieces to recreate the original picture. In our puzzle task, the original picture is rectangular and divided into N smaller rectangular pieces, each labeled based on its position in the picture. At the beginning, the puzzle pieces are randomly placed on the computer screen. While they can be moved, rotation is not allowed. When two neighboring pieces are correctly placed next to each other, they merge to form a larger piece and become inseparable. Solving the puzzle entails finding the unique configuration of these N pieces on the two-dimensional array. Our approach exclusively emphasizes the color factor during puzzle solving. The main objective is to comprehend the importance of human pattern recognition in puzzle solving and replicate the problem-solving strategies used by human puzzle solvers.

Related work

In 1964, Freeman and Gardner proposed the first jigsaw solver for apictorial 9 piece puzzles, focusing solely on the shape of the pieces [24]. Their method, called partial boundary curve matching, identifies critical points along the edges and calculates how well the pieces fit together. Radack and Badler also attempted puzzle solving using partial boundary curve matching with polar coordinates [25].

Apart from curve matching methods, the image on puzzle pieces plays a significant role in puzzle solving, especially for human solvers. In 1994, Kosiba et al. developed the first algorithm that utilized both image and shape information of puzzle pieces to solve puzzles with up to 54 pieces [26]. Other authors have also proposed similar algorithms based on both shape and image, where they first assemble the frame pieces and then employ a greedy algorithm to fill in the interior [27, 28]. These algorithms can handle puzzles with several hundred pieces, but the reconstruction results are usually reported for only one or a few images.

There has been a growing interest in pictorial puzzles with pieces of rectangular shape [18, 29, 30]. In this case, a puzzle solver utilizes the pictorial information on the pieces to construct the original picture by correctly assembling all pieces. Such a solver typically consists of two main modules: a compatibility metric that uses a cost function to evaluate the likelihood of a given pair of pieces being neighbors in the original configuration, and an assembly algorithm that determines the placement of pieces according to the compatibility metric. Cho and coworkers discussed several interesting applications of the rectangular piece puzzle in image editing and synthesis.

Methods

In Sect. "A nucleation model of puzzle solving", we introduce a puzzle-solving nucleation model as a key component of our effort to create automated solvers that mimic human behavior. We examine its time complexity, a crucial aspect of our research. This model emulates the human approach of commencing with a piece and gradually expanding it during the puzzle-solving process. Moving to Sect. "Statistical analysis of edges", we delve into the attributes of puzzle pieces serving as nucleation sites. Our analysis draws on data from an empirical database, detailed in the Supplementary Information, encompassing 8 puzzle pictures displayed in Fig. 1. By combining insights from Sect. "A nucleation model of puzzle solving" and "Statistical analysis of edges", we present an automated puzzle solver in Sect. "Computer

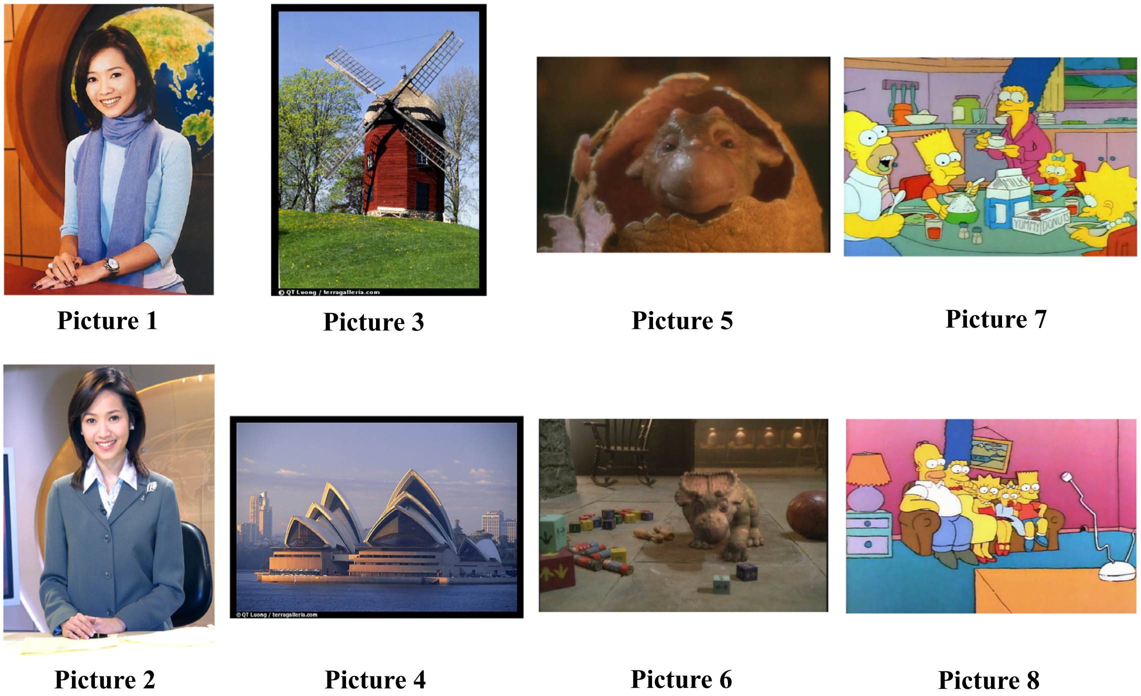


Fig. 1 Eight different pictures used to examine human perception of color patterns in puzzle solving. Among these pictures, two (pictures 1 and 2) are portraits, two (pictures 3 and 4) are buildings, two (pictures 5 and 6) are animals, and two (pictures 7 and 8) are cartoons

simulation algorithm", specifically crafted to replicate the general behaviors demonstrated by human solvers. Figure 2 illustrates the framework of this study.

A nucleation model of puzzle solving

In a puzzle composed of N non-rotatable rectangular pieces, there exist $N!$ potential arrangements when assigning each piece to a position. As a result, employing a brute force algorithm to solve the puzzle would demand O($N!$) operations.

Alternatively, we can approach puzzle-solving using a nucleation process combined with a trial and error approach. Initially, we select an arbitrary puzzle piece to serve as the nucleation site and then search for its neighboring piece with the corresponding edge from the remaining N–1 pieces. Once a correct choice is made, these two pieces are merged to form a larger piece. Subsequently, there are N–2 choices from the remaining pieces to fit one of the new edges for the merged nucleation site. The overall number of trials required to solve the puzzle is $(N–1) + (N–2) + \cdots + 1$, which simplifies to $N(N–1)/2$, leading to a time complexity of the order N^2. Notably, in this approach, puzzle-solving is achieved without utilizing any color information on the puzzle pieces. However, if color information is employed to assist in puzzle-solving, the number of trials needed to solve the puzzle would likely be reduced. In the general, we anticipate a power law relationship between the number of trials (T) and the number of puzzle pieces (N), denoted by $T \propto N^{\lambda}$, where $\lambda < 2$. The value of λ depends on the available color information on the puzzle pieces.

Statistical analysis of edges

In computer simulations aimed at replicating human puzzle-solving behavior, the statistical analysis of edge features assumes a significant role. Specifically, features such as distinctive and iridescent edges in puzzle pieces are crucial for mimicking human solvers effectively. Empirical evidence indicates that during the initial stages of puzzle solving, human solvers tend to select pieces that are less common and exhibit greater color variation. These characteristics are particularly important for successful edge matching.

To analyze the color pattern of a puzzle piece, as demonstrated in Fig. 3a, we divide it into k^2 sections, with each edge containing k sections. In contrast to the RGB color space, where most human solvers do not differentiate between similar colors during the puzzle-solving process, we employ a set of 16 colors (as shown in Fig. 3b) to represent color pixels within each section. This alternative color representation allows us to symbolize color patterns effectively.

To facilitate our analysis, we define the set of edges as $E = \{$top-edge (1), bottom-edge (–1), left-edge (2), right-edge (–2)$\}$. We refer to the edge e (where e belongs to E) of piece a and the edge -e of piece b as relative edges ($a \neq b$) and define the level of their resemblance as

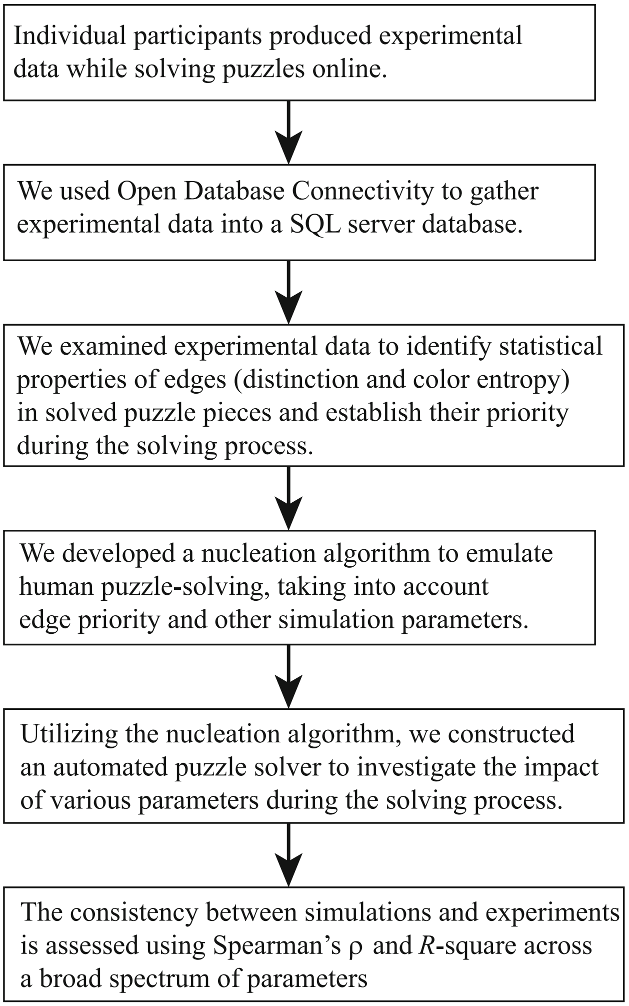

Individual participants produced experimental data while solving puzzles online.
We used Open Database Connectivity to gather experimental data into a SQL server database.
We examined experimental data to identify statistical properties of edges (distinction and color entropy) in solved puzzle pieces and establish their priority during the solving process.
We developed a nucleation algorithm to emulate human puzzle-solving, taking into account edge priority and other simulation parameters.
Utilizing the nucleation algorithm, we constructed an automated puzzle solver to investigate the impact of various parameters during the solving process.
The consistency between simulations and experiments is assessed using Spearman's ρ and R-square across a broad spectrum of parameters

Fig. 2 An outline of the framework for puzzle-solving utilizing knowledge-based automation

$$R^{a,b;e} \equiv \frac{1}{k}\sum_{i=1}^{k}\frac{\vec{C}^{\,a;e}\cdot\vec{C}^{\,b;-e}}{\left|\vec{C}^{\,a;e}\right|\left|\vec{C}^{\,b;-e}\right|} \tag{1}$$

where $\vec{C}^{\,a;e}$, calculated by tallying the number of pixels for each color, denotes the color vector of the i-th section of an edge x in piece a. $R^{a,b;e}$ represents the degree of similarity between two relative edges, indicating the likelihood of a human solver attempting to match pieces (a, b) with the edges (e, -e) during puzzle solving. Due to the imprecision of human perception regarding color distribution on edges, if $R^{a,b;e}$ exceeds a threshold value R_t, the color patterns of two edges are considered 'similar". The ambiguity of finding a corresponding edge increases as the number of similar edges for a particular edge grows. To quantify this, we introduce the concept of the percentage of similar edges for edge e of piece a, denoted as $P^{a;e} \equiv M^{a;e}/N - 1$, where $M^{a;e}$ represents the count of similar edges and N is the number of puzzle pieces. Additionally, we define the color variation of an edge (e.g., edge e of piece a) using its color entropy as follows:

$$S^{a,e} \equiv -\sum_{i=1}^{k}\sum_{C=1}^{16} p_{i,C}^{a;e}\log\left(p_{i,C}^{a;e}\right), \tag{2}$$

where $p_{i,C}^{a;e}$ is the probability distribution of color C in section i of edge e. An edge with a higher value of $S^{a;e}$ exhibits more iridescence, while an edge with $S^{a;e} = 0$ is monochromatic.

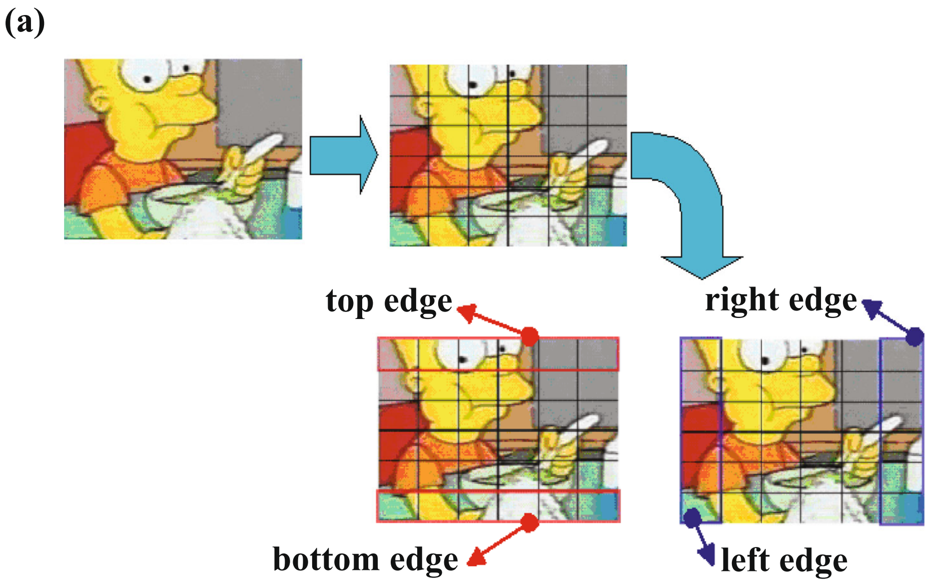
Our analysis of an empirical database, as presented in the supplementary information, reveals that human solvers solve puzzles effectively by preferentially selecting puzzle pieces with distinctive (small $P^{a;e}$) and iridescent (large $S^{a;e}$) edges. Notably, it is observed that human solvers commonly initiate the puzzle-solving process by focusing on pieces with distinctive edges ($\mathrm{E}_1 = \{\text{edges with } P^{a;e} \le 0.22\}$) and iridescent edges ($\mathrm{E}_2 = \{\text{edges with } S^{a;e} > 2.3\}$). Among these two characteristics, distinctive edges appear to have a more significant impact on their selection strategy than iridescent edges. As the puzzle-solving progresses, they gradually enlarge the initially chosen piece. Based on the above observations, we categorize the edges of puzzle pieces into three types: {A}, $\mathrm{E}_1 \cap \mathrm{E}_2$; {B}, $\mathrm{E}_1 - \mathrm{E}_2$; and {C}, otherwise. In a typical puzzle-solving process, it is reasonable for us to assume that the likelihood of being a nucleation site follows the order: $P(\{\mathrm{A}\}) > P(\{\mathrm{B}\}) > P(\{\mathrm{C}\})$.

To validate our assumption, we employed data mining techniques, specifically association rules (X → Y), to calculate the support and confidence for edges belonging to the three sets to appear at the first stage of the puzzle-solving process [31]. In this context, X represents possible edges of type X = {A, B, or C} found in the empirical database, while Y represents the solved edges observed at the first stage. We calculated the support(X → Y) as σ(edges of type X at the first stage)/σ(edges of all types in the database) and the confidence(X → Y) as σ(edges of type X at the first stage)/σ(edges of type X in the database), where σ is the count of events. Indeed, the results in Table 1 suggest that edges of type {A} are more likely to appear at the first stage of puzzle solving than those of type {B} or type {C}.

Computer simulation algorithm

To simulate the human puzzle-solving process, as illustrated in the flow chart of Fig. 4, we developed an automated algorithm to complete the puzzle task starting from a nucleation site. This algorithm functions as follows by specifying the picture ID, the number of puzzle pieces, and the parameter values (R_t, α, k, p_A, p_B):

Fig. 3 **a** Partition of a puzzle piece into 36 sections. Each piece contains 4 edges (top, bottom, left, and right), and each edge consists of 6 sections. **b** The RGB definition of 16 colors used to specify color patterns



(b)

Quan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	0	0	0	0	0	0	0	128	128	128	128	192	255	255	255	255
G	0	0	0	128	128	255	255	0	0	128	128	192	0	0	255	255
B	0	128	255	0	128	0	255	0	128	0	128	192	0	255	0	255

Table 1 Association rule (X → Y) as well as the support and confidence for edges in sets {A}, {B}, and {C} to appear at the first stage of puzzle solving process

X	Y	support	confidence
Edges of type {A}	Edges appear at the first stage	0.08	0.36
Edges of type {B}	Edges appear at the first stage	0.06	0.21
Edges of type {C}	Edges appear at the first stage	0.03	0.07

1. The algorithm selects a picture from those in Fig. 1 and divides it into N rectangular pieces.
2. For each edge, the algorithm calculates its value of $P^{a;e}$ and $S^{a;e}$.
3. Edges are categorized into three sets labeled as {A}, {B}, and {C} based on their $P^{a;e}$ and $S^{a;e}$ attributes.
4. To start solving, the algorithm picks a nucleation edge with probabilities p_A from {A}, p_B from {B}, and $1-p_A-p_B$ from {C}, where $p_A > p_B > 1-p_A-p_B$. This reflects human limitations in discerning picture details.
5. The algorithm creates a list for relative edges of the remaining pieces with $R^{a,b;e} \geq R_t$ and randomly selects one edge to check if it corresponds to the chosen edge. If no match is found, the selected edge is removed from the list in subsequent attempts.
6. When a pair of corresponding edges is matched, they are merged to form a larger piece, and the nucleation site expands as more corresponding edges are found during the puzzle-solving process.
7. The algorithm continues the process, selecting one edge of the new piece based on $P^{a;e}$ and $S^{a;e}$ to find its corresponding edge. If none of the selected edges fit, the algorithm selects another nucleation edge from the nucleation site.
8. The above procedures continue until the puzzle is solved, and the total number of attempts to solve the puzzle (T) is recorded.
9. Towards the late stage of puzzle-solving, when an empty list of relative edges (i.e., $R^{a,b;e} < R_t \ \forall b$) is encountered, the algorithm reduces the threshold value to αR_t, where $\alpha < 1$.

Figure 5 depicts a snapshot of the puzzle-solving process using picture 8. Throughout the study, puzzles using the eight pictures in Fig. 1 and the number of pieces N between 4 and 100 were considered. 30 computer simulations were performed for each case, and all puzzle-solving processes were recorded in a simulation database.

Results and discussion

In this study, we employed eight different pictures from Fig. 1 for an automated puzzle solver to model human recognition of color patterns [32]. These pictures included two portraits (pictures 1 and 2), two buildings (pictures 3 and 4), two animals (pictures 5 and 6), and two cartoons (pictures 7 and 8). Among the pictures used, pictures 6 and 7 stand out due to their inclusion of multiple objects, providing more opportunities for multiple nucleation compared to pictures that feature a single object. Unlike human solvers, who are capable of employing multiple nucleation strategies to solve puzzles, the current automated solver relied exclusively on a single nucleation approach for puzzle-solving. In this study, we will compare our simulation results with those from both the dataset of 8 pictures and the dataset of 6 pictures (excluding pictures 6 and 7).

Our algorithm utilizes a nucleation rule for puzzle solving, as illustrated in Fig. 4, which effectively simulates the solving process employed by human solvers. Unlike brute force algorithms with an $N!$ dependence, our approach shows a power-law relation between the average solving time (T) and the number of pieces (N), denoted as $T(N) \propto N^{\lambda}$. The value of λ quantifies the efficiency in processing pictorial information, with smaller values implying better puzzle-solving efficiency.

For monochromatic puzzles lacking useful color information (Figure S3), the measured value of λ is approximately 1.85 based on trial and error. However, for puzzle pictures with specific color patterns, as shown in Fig. 6, the values of λ fall within the range of 1.60 and 1.77 when using the parameter set $k = 6$, $R_t = 0.65$, $\alpha = 0.55$, and $\{p_A, p_B, p_C\} = \{94, 4, 2\}$ (as percentages). The observed power-law behavior in our simulations closely resembles the puzzle-solving patterns of human solvers in the empirical database, where the exponent λ_H ranges from 1.48 to 1.67. The inset of Fig. 6 demonstrates linear regressions of the data points of (λ, λ_H) from the empirical database and computer simulations with zero intercept, yielding $\lambda = 1.06 \cdot \lambda_H$ with $R^2 = 0.79$ for the dataset of 8 pictures and $\lambda = 1.05 \cdot \lambda_H$ with $R^2 = 0.85$ for the dataset of 6 pictures. This finding indicates that our simulation algorithm successfully mimics human solvers in puzzle solving.

In general, pictorial information within puzzle pieces can provide valuable clues to solve the puzzle. The small values of λ for pictures 6 and 7 are attributed to the fact that both pictures contain multiple objects, which act as helpful hints during the puzzle-solving process. On the other hand, pictures 5 and 8 have large values of λ because a significant portion of the picture exhibits a monotonic color distribution. It's important to emphasize that our computer simulations exclusively focused on solving processes with a single nucleation site. In contrast, for pictures 6 and 7, human solvers may initiate multiple nucleation sites during the solving process. This leads to the values of λ_H (1.48 and 1.48) being smaller than the corresponding λ (1.61 and 1.60) for these pictures. This observation raises an interesting topic for our future investigation: exploring the performance of puzzle solving with different strategies.

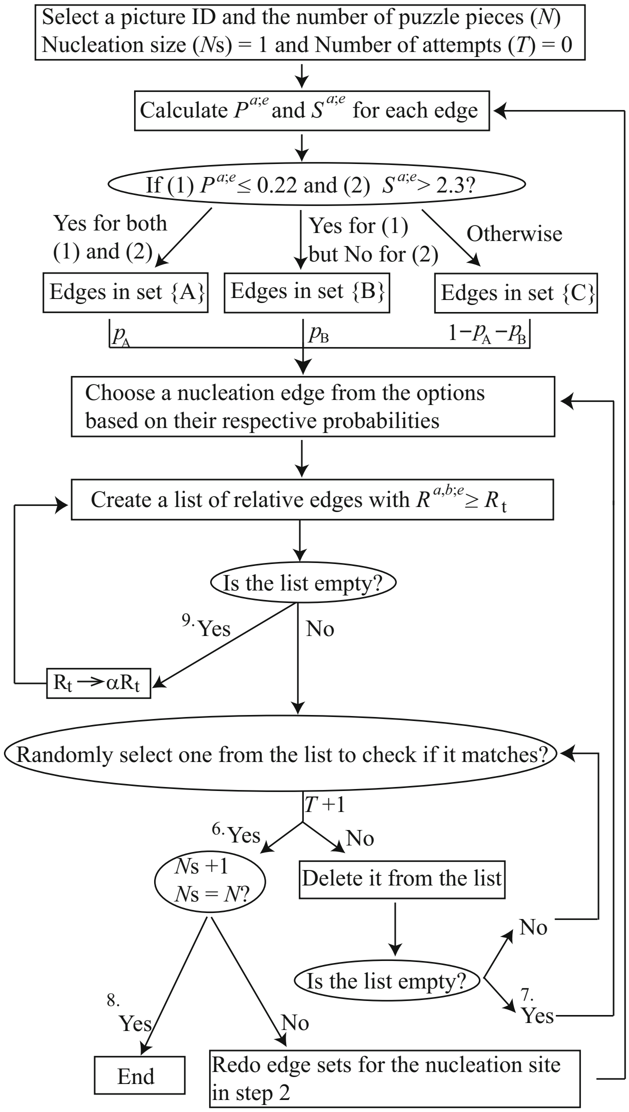

Select a picture ID and the number of puzzle pieces (N)
Nucleation size (Ns) = 1 and Number of attempts (T) = 0
Calculate P^{a;e} and S^{a;e} for each edge
If (1) P^{a;e} ≤ 0.22 and (2) S^{a;e} > 2.3?
Yes for both (1) and (2)
Yes for (1) but No for (2)
Otherwise
Edges in set {A}
Edges in set {B}
Edges in set {C}
p_A
p_B
1−p_A−p_B
Choose a nucleation edge from the options based on their respective probabilities
Create a list of relative edges with R^{a,b;e} ≥ R_t
Is the list empty?
9. Yes
No
R_t → αR_t
Randomly select one from the list to check if it matches?
T +1
6. Yes
No
Ns +1
Ns = N?
Delete it from the list
No
Is the list empty?
7. Yes
8. Yes
No
End
Redo edge sets for the nucleation site in step 2

Fig. 4 Flow chart of the nucleation algorithm of the automated puzzle solver

Figure 7 depicts the collapsed-data diagram of our automated solver for both datasets, comprising 8 pictures and 6 pictures. The horizontal axis depicts Spearman's ρ value, indicating the strength and direction of the monotonic relationship between our nucleation algorithm's outcomes and the experiment's observations. The vertical axis illustrates the R-squared value, assessing how well the model fits the observed data. Combining both measures offers a more thorough evaluation, capturing various aspects of the relationship

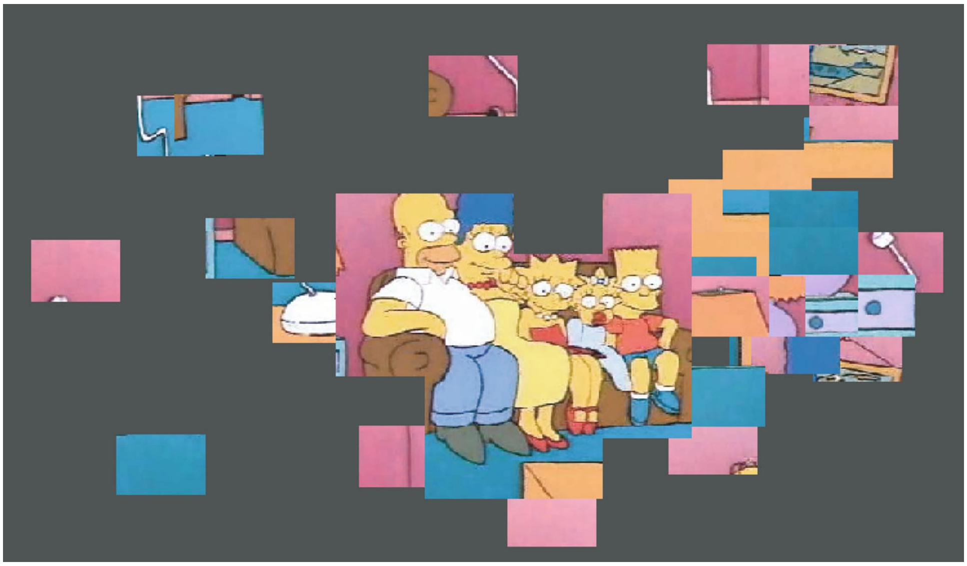


Fig. 5 Snapshot of the puzzle solving process with the automated solver using the nucleation algorithm

	Picture 1	Picture 2	Picture 3	Picture 4	Picture 5	Picture 6	Picture 7	Picture 8
λ_H	1.54	1.58	1.65	1.57	1.67	1.48	1.48	1.66
λ	1.60	1.61	1.74	1.68	1.77	1.61	1.60	1.75

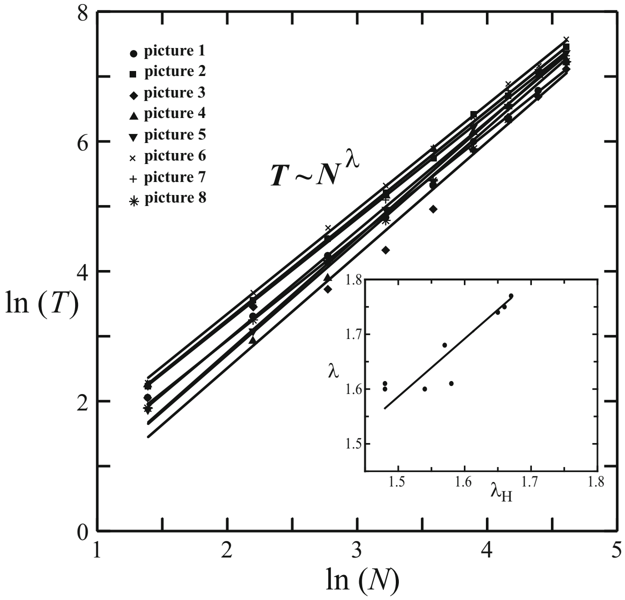


Fig. 6 Average puzzle solving time as a function of N. A power-law form with an exponent between 1.4 and 1.8 is found. The inset shows an excellent agreement between the scaling exponents calculated from simulations and their empirical values

between variables and overcoming individual measure limitations. Data collection spanned a wide range within the parameter space, with emphasis on $k = 6$, $R_t = 0.65$, $\alpha = 0.55$, and $\{p_A, p_B, p_C\} = \{90, 8, 2\}$ (depicted by dashed circles). Theoretically, for closely mimicking human solving patterns, the optimal parameter range aligns with the upper-right quadrant of Fig. 7, highlighting a pronounced effect size. [33] Overall, our automated solver exhibits a more pronounced effect size when applied to the dataset containing 6 pictures, which is predominantly influenced by a single nucleation center. In this scenario, a stronger correlation between λ and λ_H is observed, particularly with higher p_A values. Furthermore, we noted an exceptionally robust correlation with $\rho = 1.0$ and $R^2 = 0.94$ when slightly adjusting the k value in accordance with N ($k = 7$ for $N = 2, 3$, or 4, $k = 6$ for $N = 5, 6$, or 7, and $k = 5$ for $N = 8, 9$, and 10), indicated as $k(N)$. This adjustment is driven by the reduction in color pixels within puzzle pieces as N increases. Figure 7 illustrates the significant sensitivity of our results to the parameters R_t and α, with the correlation substantially diminishing as the parameter values (R_t, α) move away from (0.65, 0.55). Notably intriguing is the observation that our simulations, conducted with large R_t values ($R_t > 0.7$), exhibit a diminished correlation with experimental results, primarily attributed to relatively small λ values in comparison to λ_H. This discrepancy is logical, as our automated solver demonstrates superior proficiency in discerning color patterns compared to humans, especially in scenarios involving large R_t. From a computational perspective, the solver proves significantly more efficient than humans in puzzle-solving, leading to a weaker consistency between simulations and experimental outcomes.

The process of puzzle solving, reminiscent of the endeavor to find a specific configuration within a vast configuration space, bears resemblances to protein folding. Levinthal underscored the impracticality of an unfolded protein navigating the entire conformation space at random and proposed the existence of a distinct pathway to its native state. To observe this phenomenon, we meticulously scrutinized all puzzle-solving procedures, documenting the likelihood of successfully solved puzzle pieces at each stage. Within this framework, the puzzle-solving process is segmented into five

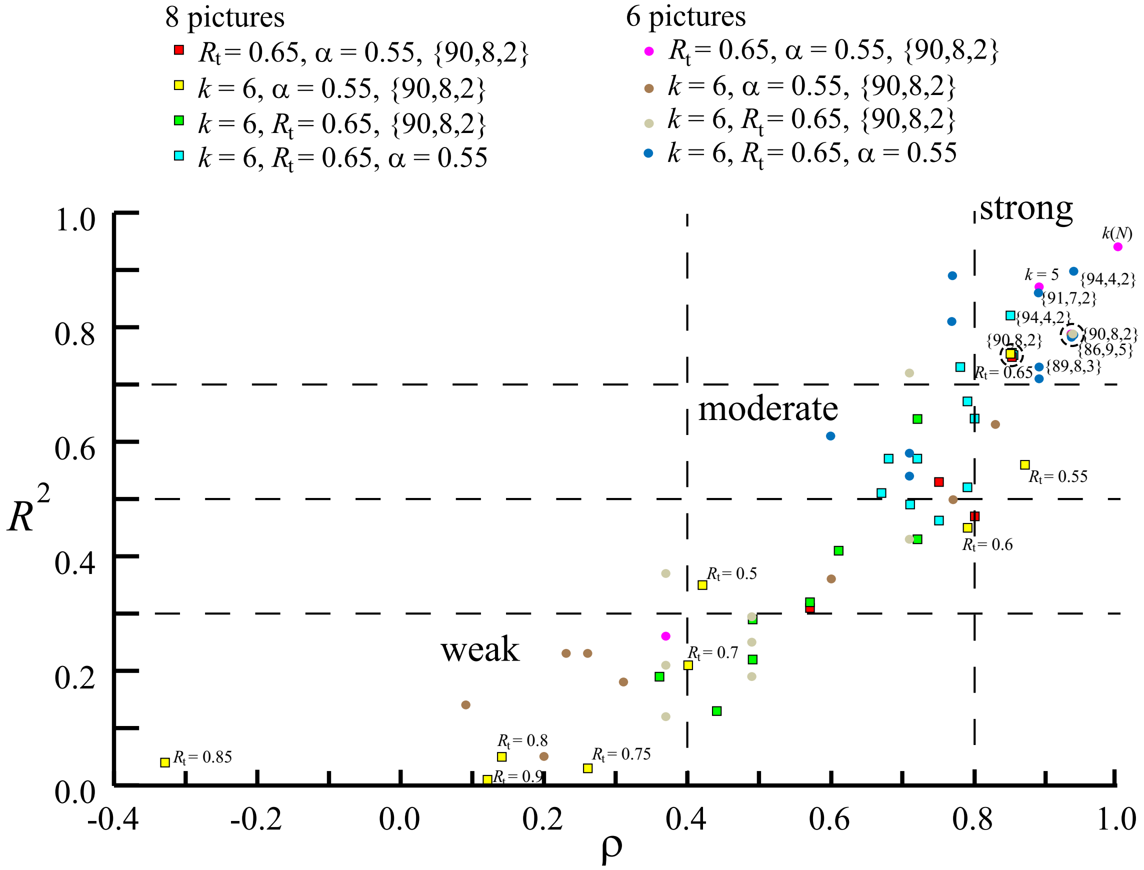


Fig. 7 A collapsed-data diagram of the automated solver for both datasets, comprising 8 pictures and 6 pictures. The horizontal axis corresponds to the Spearman's ρ, while the vertical axis represents the R-squared value. Here $k(N)$ signifies a slightly modified k value according to N

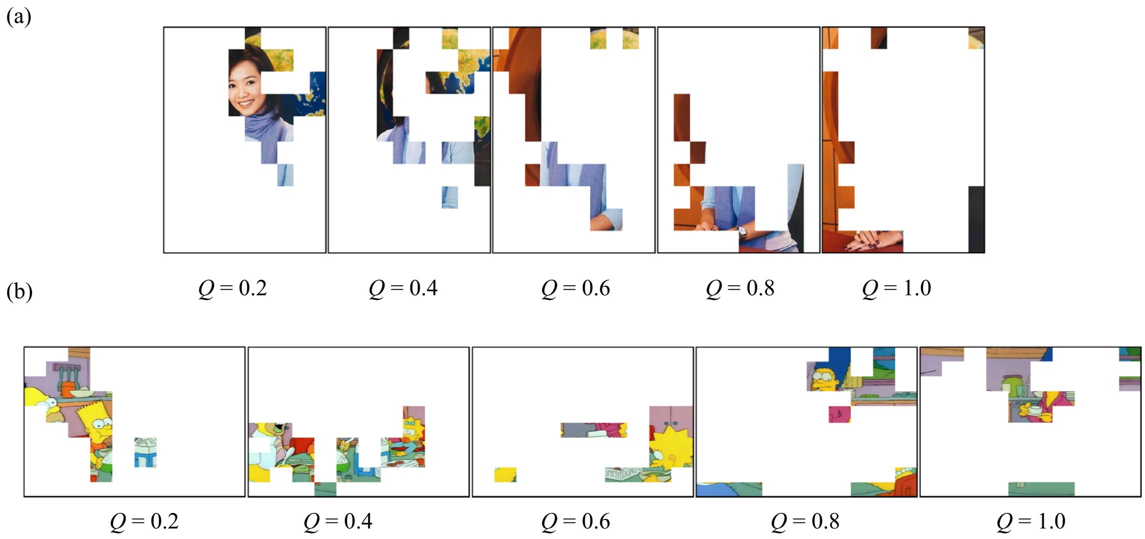


Fig. 8 Principal trails to the native configuration for two 10 × 10 puzzles: **a** picture 1 and **b** picture 8. At each stage, only puzzle pieces having greater than 26% probability of being solved during the stage are displayed

equally distributed stages, each marked by the native contact value (i.e., the percentage of completion) denoted as Q. This study also unveiled prevalent pathways in puzzle solving, as illustrated in Fig. 8. In this figure, we present puzzle pieces from pictures 1 and 7, indicating a likelihood of being solved at a stage surpassing 26% for each of the five stages. In the case of picture 1 in Fig. 8a, the facial section is recognized at the earliest stage ($Q = 0.2$) and functions as a nucleation center for identifying the complete portrait. The background surrounding the portrait in picture 1 features a uniform color palette, leading to its resolution in subsequent stages. Similarly, within Fig. 8b, the cartoon characters within picture 7 are identified during the initial stage, followed by the subsequent recognition of the uniform background. These central pathways leading to puzzle solving are distinctly evident across all eight pictures in Supplementary Figures S4 and S5. Remarkably, the color pattern on puzzle pieces plays a pivotal role in determining the trajectory toward the ultimate solution. This observation aligns with the experimental findings derived from the empirical database, as showcased in Figure S6. However, as depicted in Figures S7 and S8, the process of solving pictures 1 and 7 diverges in terms of the number of potential nucleation centers. In contrast to picture 1, picture 7 incorporates multiple nucleation centers, resulting in several primary pathways to puzzle solving.

Conclusions

In this study, we have introduced a nucleation algorithm for puzzle solving and utilized computer simulations to emulate the puzzle-solving processes employed by human solvers. Based on this study of the empirical puzzle-solving database, we observed that puzzle solving of humans can be effectively explained through a nucleation model, where puzzle pieces with distinctive (small $P^{a;e}$) and iridescent (large $S^{a;e}$) edges are commonly chosen as the nucleation site.

Our proposed automated solver assesses the efficiency of human solvers in puzzle solving by adopting the nucleation strategy and considering the pictorial information on puzzle pieces. The average puzzle solving time follows a power-law relationship with N, with an exponent less than 2. The specific value of this exponent is influenced by the available pictorial information in the puzzle picture. Interestingly, we speculate on the similarity between the observed efficiency in finding the puzzle solution from the vast number of configurations and that of proteins folding into a unique native structure while exploring the conformation space. Drawing inspiration from Levinthal's paradox in the protein folding problem, we identify the principal pathways for a single nucleation center or multiple nucleation centers in puzzle solving.

In summary, we statistically determined the preferential selection of puzzle pieces based on their edge features and introduced a nucleation algorithm to emulate the puzzle-solving strategies used by humans. This approach notably enhances the efficiency of solving jig swap puzzles by leveraging pictorial information from puzzle images. Given the absence of shape information for individual pieces, this problem formulation poses even greater challenges than conventional jigsaw puzzles. Our study makes a valuable contribution to developing an architectural model for understanding human recognition of color patterns and its potential applications in problem-solving.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s40747-024-01440-0.

Author Contributions S. Fauzia and S. Chen contributed equally. S. Chen was responsible for Methodology and Software development. S. Fauzia handled Visualization, Investigation, and Formal Analysis. R.-J. Hsu focused on Data Curation. R. Chen carried out Formal Analysis and Writing - Reviewing and Editing. C.-M. Chen was in charge of Conceptualization, Writing - Original Draft Preparation, Supervision, Project Administration, and Funding Acquisition.

Funding This work funded by National Science and Technology Council to Chi-Ming Chen with Grant number MOST 111-2112-M-003-009.

Data availability Data are located in controlled access data storage at National Taiwan Normal University, and are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical and informed consent for data used Informed consent was obtained from all individual participants included in the study. No personal data was collected during the course of the experiment.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Silver D et al (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529(7587):484–489. https://doi.org/10.1038/nature16961
2. Altman T (1990) Solving the jigsaw puzzle problem in linear time. Appl Artif Intell 3(4):453–462. https://doi.org/10.1080/08839518908949937

3. Finkelstein L, Markovitch S (1998) A selective macro-learning algorithm and its application to the NxN sliding-tile puzzle. J Artif Intell Res 8:223–263. https://doi.org/10.1613/jair.484
4. Kaur A, Chauhan APS, Aggarwal AK (2019) Machine learning based comparative analysis of methods for enhancer prediction in genomic data. In: 2019 2nd International Conference on intelligent communication and computational techniques (ICCT), 28–29 Sept. 2019, pp 142–145, https://doi.org/10.1109/ICCT46177.2019.8969054.
5. Littman ML, Keim GA, Shazeer N (2002) A probabilistic approach to solving crossword puzzles. Artif Intell 134(1–2):23–55. https://doi.org/10.1016/S0004-3702(01)00114-X
6. Maini DS, Aggarwal AK (2018) Camera position estimation using 2D image dataset. Int J Innov Eng Technol 10(2):199–203
7. Xiao J, Aggarwal A, Rage U, Katiyar V, Avtar R (2023) Deep learning-based spatiotemporal fusion of unmanned aerial vehicle and satellite reflectance images for crop monitoring. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3297513
8. Peng Z, Song X, Shuai S, Stojanovic V (2023) Hysteresis quantified control for switched reaction–diffusion systems and its application. Complex Intell Syst. https://doi.org/10.1007/s40747-023-01135-y
9. Song X, Wu N, Shuai S, Stojanovic V (2023) Switching-like event-triggered state estimation for reaction-diffusion neural networks against DoS attacks. Neural Process Lett 55:1–22. https://doi.org/10.1007/s11063-023-11189-1
10. Song X, Wu N, Song S, Zhang Y, Stojanovic V (2023) Bipartite synchronization for cooperative-competitive neural networks with reaction–diffusion terms via dual event-triggered mechanism. Neurocomputing 550:126498. https://doi.org/10.1016/j.neucom.2023.126498
11. Wei T, Li X, Stojanovic V (2021) Input-to-state stability of impulsive reaction–diffusion neural networks with infinite distributed delays. Nonlinear Dyn. https://doi.org/10.1007/s11071-021-06208-6
12. Duch W, Mandziuk J (2009) Challenges for computational intelligence (Duch, W. and Mandziuk, J., Eds.; 2007) [Book reviews]. IEEE Trans Neural Netw 20(3):542–543. https://doi.org/10.1109/TNN.2009.2015975
13. Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y (2004) An integrated theory of the mind. Psychol Rev 111(4):1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036
14. Laird JE, Newell A, Rosenbloom PS (1987) SOAR: an architecture for general intelligence. Artif Intell 33(1):1–64. https://doi.org/10.1016/0004-3702(87)90050-6
15. Carbonell J et al (1991) PRODIGY: an integrated architecture for planning and learning. SIGART Bull 2(4):51–55. https://doi.org/10.1145/122344.122353
16. Demaine ED, Demaine ML (2007) Jigsaw puzzles, edge matching, and polyomino packing: connections and complexity. Graphs Combin 23(1):195–208. https://doi.org/10.1007/s00373-007-0713-4
17. Zhao YX, Su M-C, Chou ZL, Lee J (2007) A puzzle-solver-based speech descrambling method. WSEAS Trans Comput 6:725–732
18. Cho TS, Avidan S, Freeman WT (2010) A probabilistic image jigsaw puzzle solver. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 13–18 June 2010, pp 183–190, https://doi.org/10.1109/CVPR.2010.5540212.
19. Caggiano S, De Marsico M, Distasi R, Riccio D (2015) Multi-Object segmentation for assisted image reconstruction. In: International Conference on pattern recognition applications and methods, Lisbon, Portugal, 2015, vol. 2, pp. 100-107, doi: https://doi.org/10.5220/0005274601000107
20. Derech N, Tal A, Shimshoni I (2021) Solving archaeological puzzles. Pattern Recognit 119:108065. https://doi.org/10.1016/j.patcog.2021.108065
21. Lin H-Y, Fan-Chiang W-C (2012) Reconstruction of shredded document based on image feature matching. Expert Syst Appl 39(3):3324–3332. https://doi.org/10.1016/j.eswa.2011.09.019
22. Marande W, Burger G (2007) Mitochondrial DNA as a Genomic Jigsaw Puzzle. Science (New York N.Y.) 318:415. https://doi.org/10.1126/science.1148033
23. Helmer-Citterich M, Tramontano A (1994) PUZZLE: A new method for automated protein docking based on surface shape complementarity. J Mol Biol 235(3):1021–1031. https://doi.org/10.1006/jmbi.1994.1054
24. Freeman H, Gardner L (1964) A pictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. IEEE Trans Electron Comput 13:118–127. https://doi.org/10.1109/PGEC.1964.263781
25. Radack GM, Badler NI (1982) Jigsaw puzzle matching using a boundary-centered polar encoding. Comput Graph Image Process 19(1):1–17. https://doi.org/10.1016/0146-664X(82)90111-3
26. Kosiba DA, Devaux PM, Balasubramanian S, Gandhi TL, Kasturi K (1994) An automatic jigsaw puzzle solver. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision & Image Processing, 9–13 Oct 1994, vol. 1, pp 616–618. https://doi.org/10.1109/ICPR.1994.576377.
27. Makridis M, Papamarkos N (2006) A New Technique for Solving a Jigsaw Puzzle. In: 2006 International Conference on image processing, 8–11 Oct. 2006, pp 2001–2004, https://doi.org/10.1109/ICIP.2006.312891.
28. Nielsen TR, Drewsen P, Hansen K (2008) Solving jigsaw puzzles using image features. Pattern Recognit Lett 29(14):1924–1933. https://doi.org/10.1016/j.patrec.2008.05.027
29. Alajlan N (2009) Solving square jigsaw puzzles using dynamic programming and the hungarian procedure. Am J Appl Sci 6(11):1941–1947. https://doi.org/10.3844/ajassp.2009.1941.1947
30. Gallagher AC (2012) Jigsaw puzzles with pieces of unknown orientation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 16–21 June 2012, pp 382–389, https://doi.org/10.1109/CVPR.2012.6247699.
31. Tan P-N, Steinbach M, Kumar V (2014) Introduction to data mining. Pearson, New York
32. Rösler F, Heil M, Hennighausen E (1995) Distinct cortical activation patterns during long-term memory retrieval of verbal, spatial, and color information. J Cogn Neurosci 7:51–65. https://doi.org/10.1162/jocn.1995.7.1.51
33. Rea LM, Parker RA (1997) Designing and conducting survey research: a comprehensive guide. Josey-Bass Publishers, San Francisco

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.