The Industry of the Future: From Industry 4.0 to Industry 5.0 – Integration of Humans and Technology: New Technologies

Karabegović, Isak

2025

Akademija nauka i umjetnosti Bosne i Hercegovine
https://bastina.anubih.ba/handle/123456789/837

DOI: 10.5644/PI2025.220.02

Cognitive and Visual-Motor Robotic Manipulation in Industrial Environments: A Puzzle Assembly Paradigm

Aleksandar Rodić*[1], Aleksandar Rodić[1], Jelena Ilić[1], Natalija Dimitrijević[2], Aleksandar Milenković[1]

Abstract: *The increasing complexity of industrial processes and the shift towards automation demand intelligent, adaptive robotic systems capable of performing cognitive and visuomotor tasks in dynamic environments. This paper presents the puzzle assembly gaming task as an illustrative paradigm for the implementation of such systems in industry. The assembly of puzzles, often considered a cognitive game, mimics many characteristics found in industrial tasks: high variability, unstructured environments, the need for visual recognition, decision-making, trajectory planning and fine manipulation. In the paper it is explored how artificial intelligence (AI), computer vision, machine learning, inference systems, and robot motion planning converge to create systems capable of tackling tasks with similar cognitive and manipulative challenges in industrial settings.*

Keywords: *Industry 4.0, cognitive robots, visual motor manipulation, puzzle game 1.*

1. Introduction

Industrial robotics has evolved from executing repetitive, pre-programmed motions to performing increasingly sophisticated tasks that require perception, adaptability, and decision-making. Traditional industrial automation relied heavily on structured environments and fixed scenarios. However, modern manufacturing settings increasingly resemble dynamic environments where objects may arrive in random orientations and positions, requiring a robot to "understand" the scene before acting. This cognitive capability is paramount for applications such as flexible assembly lines, quality inspection, and adaptive sorting.

Puzzle assembly represents an ideal testbed for these capabilities. In the puzzle paradigm, the robot is exposed to an unstructured set of randomly scattered pieces and a reference image representing the completed puzzle. The robot must use vision to recognize, identify, and classify individual pieces, reason about their placement, and perform delicate manipulation to assemble the

*[1] University of Belgrade, Institute Mihajlo Pupin, Belgrade, Serbia
E-mail:aleksandar.rodic @pupin.rs
[2] University of Belgrade, Faculty of Electrical Engineering, Belgrade, Serbia

full picture. This paper explores the conceptual foundation and implementation of such systems in industrial contexts.

The central idea of this research is to develop algorithms and a methodology that can automate the human skill of solving jigsaw puzzles, a didactic game, by leveraging visual perception of the complete image and mapping human cognitive and visuo-manipulative skills onto an industrial robot. This illustrative and educational task serves as a prototype for applications in industrial environments, where the goal is to synthesize a high level of cognitive and manipulative autonomy in robots.

The ultimate objective is to achieve a degree of automation in which a robot can independently execute not only simple, predefined operations but also more complex technological tasks that require advanced cognitive intelligence — such as perception, reasoning, and learning. Solving the robotic jigsaw puzzle problem, therefore, becomes a gateway to generalizing this knowledge and skill set to similar assembly tasks within industrial contexts.

In a review of the literature, it was found only a limited number of references directly addressing robotic puzzle assembly. One particularly interesting example includes the project “Jigsaw Puzzle Robot” [1], accompanied by video demonstrations and a technical report [2, 3], in which a gantry robot and vision system were used to solve a jigsaw puzzle. However, their method focused exclusively on the analysis of contour shapes and mechanical fit, ignoring visual parameters such as color and texture. The puzzle pieces used in their study were monochromatic (white), indicating that neither color nor surface detail was considered.

In contrast, the approach in this paper is inspired by the way humans, as biological systems, solve jigsaw puzzles. Human players rely on a combination of visual cues: color spectra, texture features, and contour shapes to infer the correct positioning of puzzle pieces. Our methodology, therefore, aims to replicate this layered decision-making strategy. We argue that such a biologically inspired, multimodal approach is more comprehensive and meaningful in the context of robotics. It promotes the development of generalizable cognitive and manipulative skills that can be transferred to complex industrial tasks, such as autonomous robotic assembly, where adaptability and perceptual intelligence are essential.

Our intention is not to create a performative or exhibition-level demonstration of robotic skill, but rather to establish a structured framework for understanding and engineering generalizable skills in autonomous robotic systems. This research contributes toward bridging the gap between human cognitive behavior and robotic autonomy in real-world industrial applications.

The project Jigsaw Puzzle Robot [1]-[3] features a gantry-style robot equipped with a vision system designed to assemble jigsaw puzzles.The robot focuses on analyzing the contours of puzzle pieces, using shape-matching algorithms to determine correct placements.Notably, the system does not utilize color or texture information, and the puzzles used are monochromatic.

The CNC Jigsaw Puzzle Building Robot[2] isdeveloped at the University of Pretoria. This project involves a CNC-based robotic system capable of assembling jigsaw puzzles.The system employs computer vision techniques to detect and classify puzzle pieces, followed by a solving algorithm to determine their correct positions.The robot then physically assembles the puzzle using precise movements.

The AI-Powered Jigsaw Puzzle Solving Robot project[3]showcases an AI-driven robotic arm designed to autonomously solve jigsaw puzzles.The system integrates computer vision for piece recognition and deep learning algorithms to predict correct placements, enabling the robot to assemble the puzzle without human intervention.

The academic study [4] presents a method for assembling jigsaw puzzles without relying on pictorial information.Instead, it utilizes integral area invariants for shape matching, allowing the system to solve puzzles based solely on the geometry of the pieces.

Several video demonstrations are available showcasing robotic systems solving jigsaw puzzles [5, 6].

These projects illustrate various approaches to robotic jigsaw puzzle assembly, ranging from contour-based methods to AI-driven solutions.Our research, which emphasizes the integration of color, texture, and shape analysis inspired by human strategies, offers a more holistic approach that could enhance the generalization of robotic assembly skills in industrial applications.

2. Conceptual Framework

The conceptual foundation for applying puzzle assembly logic to industrial robotics rests on the integration of several advanced domains within artificial intelligence and robotics [7]-[16]. These include computer vision [9,11] for perception, machine learning for interpretation and adaptation [7,9], kinematic modeling for motion control [7,8], planning algorithms for sequencing tasks, and cognitive architectures for higher-level reasoning and decision-making. Together, these components enable the robot to simulate human-like problem-solving strategies in a structured, autonomous manner. The process of robotic puzzle assembly can be broken down into a sequence of interrelated cognitive and physical stages:

Perception

The process begins with perception, where the robot acquires visual data from its environment using cameras or other sensors. This includes capturing images of both the puzzle pieces and the reference image of the completed puzzle (if available). Through advanced image processing techniques such as edge detection, segmentation, and keypoint extraction, the system isolates individual pieces and extracts relevant visual features. These may include contours, color distributions, texture patterns, and relative position in space. The output of this phase is a structured digital representation of the observed puzzle elements.

Interpretation

In this phase, the robot interprets the visual data by identifying and classifying puzzle pieces. Shape descriptors (e.g., Fourier descriptors, curvature signatures) are used to characterize the contours of each piece. Texture and color features (e.g., histograms, LBP, color moments) are analyzed to match pieces that likely belong together. Machine learning models or heuristic rules may be applied to infer edge types (e.g., corner, border, or inner piece) and predict the likelihood of a correct match between adjacent pieces. This step mimics human visual reasoning, where a person intuitively evaluates both local details and global patterns when assembling puzzles.

Decision-Making

Based on the interpreted data, the system engages in decision-making to determine the next best piece to place. This involves evaluating all candidate pieces in relation to the current state of the puzzle and estimating where each piece might fit. Scoring functions based on shape compatibility, color continuity, and texture alignment guide this selection. The robot must also consider the evolving context of the puzzle layout, dynamically updating its internal model to reflect changes after each successful placement.

Planning and Execution

Once a piece and its position are selected, the robot plans and executes the physical action needed to grasp and place the piece. This involves calculating collision-free trajectories for its robotic arm and end-effector, using inverse kinematics and motion planning algorithms. Precision is crucial, as misalignment can lead to incorrect placements or physical interference. Gripper control must also be fine-tuned to avoid damaging delicate puzzle pieces while ensuring a firm grip.

Feedback and Adaptation

After each action, the robot evaluates the outcome using sensory feedback. This may include verifying the visual alignment of the piece or detecting tactile feedback from force sensors. If an error is detected—such as a misfit or a placement in the wrong orientation—the robot updates its strategy, either retrying the action with corrections or re-evaluating its previous interpretation and decisions. This continuous feedback loop enables learning and adaptation, mirroring human trial-and-error behavior.

Ultimately, this structured process serves not only as a foundation for puzzle-solving but also as a model for generalized autonomous assembly tasks in industrial environments, where perception, interpretation, planning, and adaptation are equally critical.

2.1 Visual Reasoning Methodology in Human Puzzle Assembly

When a human player approaches the task of assembling a puzzle based on a reference image, the process begins with a visual decomposition of the integral picture into perceptually salient regions. One of the primary strategies involves segmenting the image into dominant color zones—for example, the sky typically appears in a blue hue, albeit with subtle gradations and variations within the blue spectrum. This chromatic segmentation provides an initial heuristic for narrowing down the potential spatial localization of individual pieces.

Within these color zones, players identify finer perceptual cues—such as small clouds within the sky, a leaf in a green canopy, or architectural details like rooftops or window outlines. These elements correspond to texture, defined in computer vision and perceptual psychology as the spatial variation of intensity or color that forms distinguishable surface patterns. Texture plays a crucial role in differentiating between puzzle pieces that may otherwise share similar color profiles. Thus, the early stages of assembly are heavily reliant on combined color-texture analysis.

However, when the available pieces belong to a visually homogeneous region—e.g., a large portion of sky, ocean, or wall—where both the color and texture offer limited discriminative information, the player shifts cognitive strategy toward analyzing shape and contour geometry. This involves evaluating the external edges of puzzle pieces: convexities, concavities, tabs, and blanks. The geometry of each piece must then be mentally or physically tested against candidate neighbors to find a fit that not only connects mechanically but also aligns with visual continuity in the image.

This layered approach—first utilizing global color segmentation, then localized texture matching, and finally geometric contour reasoning—mirrors human strategies for perceptual disambiguation in uncertain visual contexts. Importantly, this multistage reasoning process is not strictly sequential; rather, it is dynamically adaptive. For example, a player may simultaneously consider edge shape and surface detail when a texture cue alone is ambiguous. The integration of these perceptual modalities allows the human solver to effectively reduce the search space and resolve ambiguities through successive refinement.

This cognitive methodology serves as a powerful metaphor for designing robotic vision systems tasked with visual reasoning under uncertainty. The robotic counterpart must similarly combine global appearance models (e.g., color histograms), local texture descriptors (e.g., Gabor filters, LBP, SIFT), and edge-based shape analysis (e.g., curvature descriptors, contour matching) to infer both semantic and geometric compatibility among parts within an unstructured environment (Fig. 1).

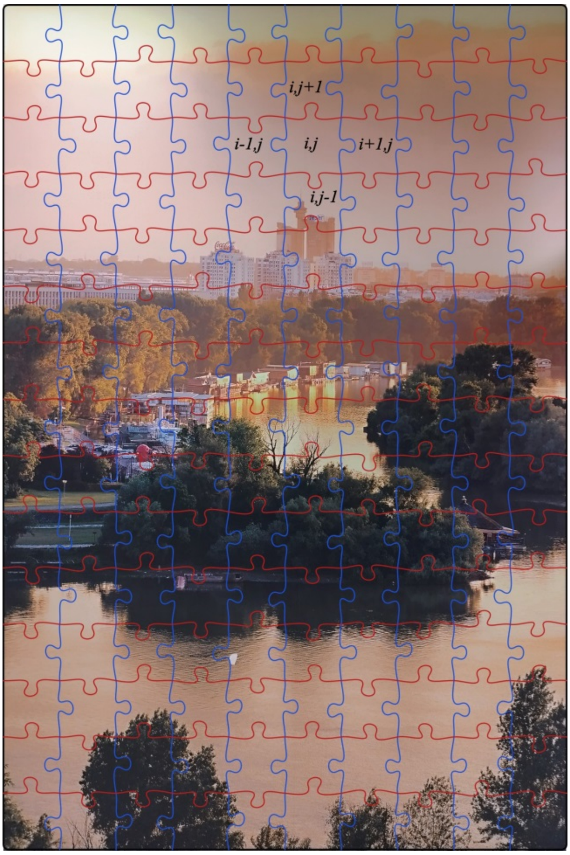


Figure 1. Arranged puzzles into a complete picture. An example of the similarity of neighboring puzzles in the sector $S_{i,j}$, $i \in \{i-1, i, i+1\}$, $j = \{j-1, j, j+1\}$ *with barely noticeable differences in color shade and texture*

2.2 Contour Comparison in Puzzle Assembly

The comparison of puzzle piece contours involves the analysis of the outer edge geometries of each piece to determine potential fits between them. This process is particularly critical in scenarios where color and texture information are insufficient for accurate identification—such as large uniform regions (e.g., sky or sea).

From a computational perspective, contour comparison is typically approached through several key steps (Fig. 2):

1. **Contour Extraction.** First, the edges of each puzzle piece are detected and extracted using image processing techniques. Algorithms such as Canny edge detection or morphological contour tracing can be applied to obtain a clean representation of the external boundaries of each piece.
2. **Segmentation into Edge Sections.** Each piece is segmented into individual edge elements—usually four in a classic jigsaw puzzle (top, bottom, left, right). Each segment is characterized either as an **inward (concave)** or **outward (convex)** shape, or as a **flat edge** in the case of border pieces.
3. **Shape Representation.** The shape of each edge is encoded using mathematical descriptors. Common representations include:
    - **Curvature descriptors**: Capture the bending of the contour line at various points.
    - **Fourier descriptors**: Transform the contour into the frequency domain for comparison.
    - **Chain codes**: Encode the direction of contour points for compact representation.
    - **Shape context descriptors**: Provide a histogram-based representation of the spatial distribution of contour points.
4. **Matching Criteria.** The goal is to match a convex edge with a corresponding concave edge such that:
    - **Geometric complementarity** is maximized (e.g., the two contours “fit” when aligned).
    - **Distance metrics** such as Hausdorff distance or sum of squared differences between contour points are minimized.
    - **Orientation alignment** is preserved, ensuring rotational consistency (especially important in robotic systems).
    - **Continuity and smoothness** are evaluated, confirming that the joint between two pieces is visually and physically seamless.
5. **Fit Scoring and Ranking**. Each candidate pair is assigned a **fit score** based on the similarity of their contours. A lower score (or higher similarity) indicates a better match. Pairs are ranked, and the top candidate is selected for further visual or mechanical validation.

In human cognition, this process is intuitive and largely subconscious. People visually inspect the "male" (tab) and "female" (blank) shapes of puzzle pieces, rotate them mentally or physically, and judge potential matches based on how well their profiles interlock. This ability is informed by experience and refined through trial and error.

In robotic systems, this contour matching process must be implemented through algorithmic pipelines that integrate vision-based contour extraction, shape analysis, and probabilistic matching models. It is especially important in unstructured environments where traditional indexing (e.g., part IDs or labels) is not available.

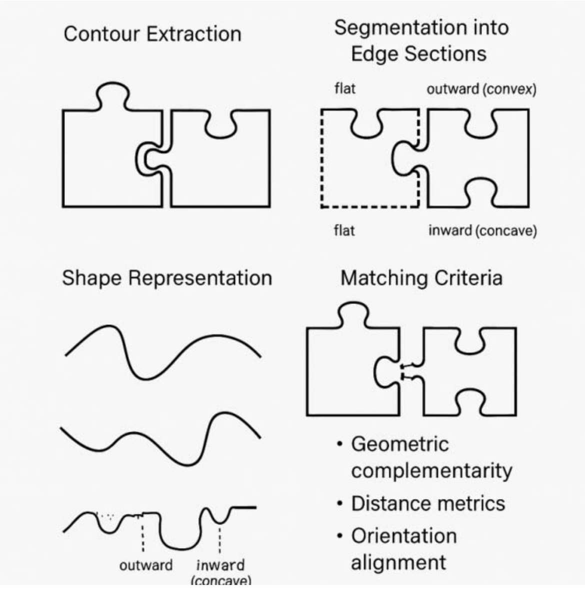


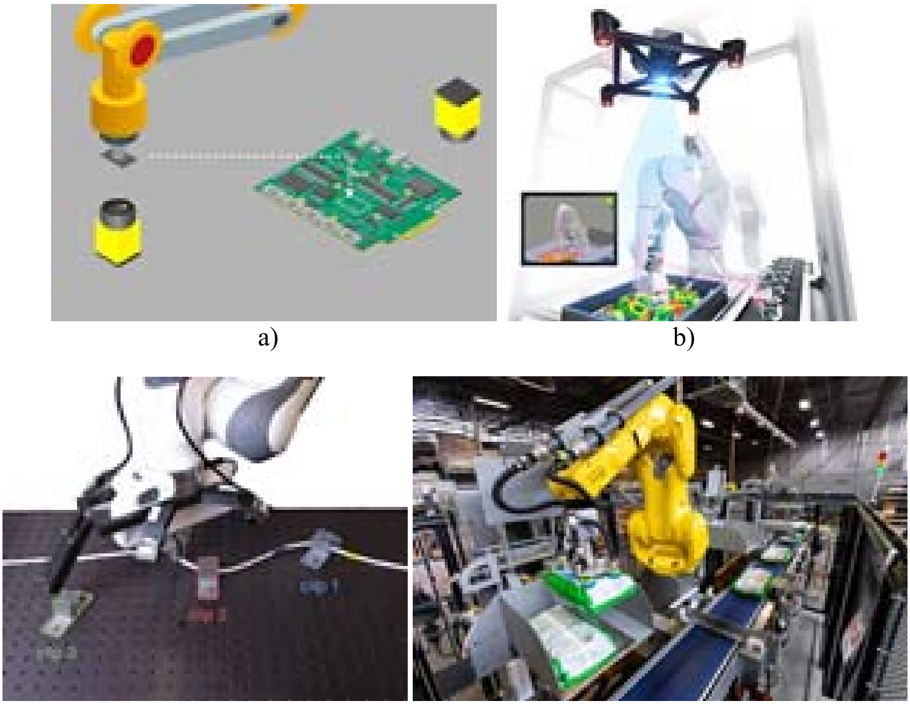
Figure 2. Puzzle contour detection

3. Industrial Analogues to Puzzle Assembly

While jigsaw puzzle assembly may initially appear as a recreational activity, its underlying principles closely mirror numerous industrial tasks that demand advanced perception, reasoning, and manipulation capabilities. By examining these parallels, we can better understand how robotic systems can be designed to handle complex assembly operations in dynamic industrial environments.

Electronic Component Placement

Automated placement of electronic components on printed circuit boards (PCBs) is a prime example (Fig. 3a). This process requires precise identification and positioning of various small components, such as resistors and integrated circuits, onto designated spots on a PCB. The task involves recognizing component types, determining their correct orientation, and placing them accurately, akin to fitting puzzle pieces based on shape and position. Advanced vision systems and robotic arms are employed to achieve the necessary precision and speed in this application.



a) b)

c) d)

Figure 3. Puzzle inspired scenarios – industrial analogues. a) PCB component placement guidance – semiconductor manufacturing and printed circuit board, b) Bin picking – 3D vision-guided robotics (Keyence), c) Cable routing system, d) Multi-line robotic packing system – Pearson packaging system

Bin Picking and Sorting

In logistics and manufacturing, bin picking involves selecting and sorting items randomly placed in containers (Fig. 3b). Robots must identify objects of varying shapes and sizes, determine their orientation, and grasp them appropriately. This scenario resembles the randomness and variability encountered in jigsaw puzzles, where each piece must be recognized and correctly oriented before placement. Implementing 3D vision systems and machine learning algorithms enables robots to handle such tasks effectively .

Flexible Assembly Tasks

Industries such as aerospace and automotive manufacturing often deal with components that arrive unsorted or require adjustments due to tolerances. Robots must adapt to these variations, making decisions on-the-fly to assemble parts correctly. This flexibility mirrors the cognitive processes involved in puzzle assembly, where each piece's placement depends on its relation to others and the overall picture. Developing robots with adaptive planning and decision-making capabilities is crucial for these applications .

Cable Routing and Hose Assembly
Routing cables and assembling hoses involve handling flexible components that require precise placement along predefined paths (Fig. 3c). Robots must recognize layout patterns, infer routing paths, and manipulate flexible parts without causing damage. This task is analogous to connecting puzzle pieces with intricate shapes and paths, demanding both visual recognition and delicate handling. Advanced control algorithms and tactile sensors are often utilized to achieve the necessary compliance and precision .

Adaptive Packaging Systems
In packaging industries, robots are tasked with arranging items of various shapes and sizes into packages efficiently (Fig. 3d). This process requires recognizing item characteristics, determining optimal placement configurations, and adapting to changing product lines. Similar to assembling a puzzle, robots must analyze visual information and make decisions to achieve the desired arrangement. Incorporating computer vision and real-time planning algorithms enables robots to perform these tasks with high efficiency .

By drawing parallels between jigsaw puzzle assembly and these industrial tasks, we highlight the importance of integrating perception, reasoning, path planing, trajectory generation and manipulation in robotic systems. Developing such capabilities is essential for achieving higher levels of autonomy and flexibility in industrial automation.

4. System Architecture

To enable a robotic system capable of solving complex tasks such as jigsaw puzzle assembly—or analogous industrial operations—a tightly integrated system architecture is required. This architecture must incorporate visual perception, learning algorithms, symbolic reasoning, motion planning, manipulation control, and a human-machine interaction interface.

Visual Perception Subsystem
Visual perception forms the first layer of information acquisition. Typically, a combination of RGB cameras and depth sensors (e.g., stereo vision, LiDAR, or Time-of-Flight sensors) is employed to capture three-dimensional

representations of the environment. These data are used for object detection, contour extraction, and spatial pose estimation. Major challenges in this subsystem include variations in lighting, partial occlusions, and diversity in object appearance regarding shape and color.

Machine Learning Models
Deep learning techniques, especially convolutional neural networks (CNNs), are applied for object classification, scene segmentation, and prediction of optimal actions. These models are trained on datasets comprising images and features of objects similar to those expected in the target scenarios. Transfer learning and fine-tuning allow the adaptation of pre-trained networks to specific real-world applications with relatively limited annotated data.

Cognitive Layer
Building upon perceptual input, the cognitive layer utilizes symbolic reasoning to decide which object to pick, where to place it, and in what sequence actions should be performed. This layer may include an inference engine employing rule-based logic, heuristics, or planning algorithms to simulate human-like decision-making processes. It is particularly crucial for orchestrating multi-step actions that require a global understanding of task objectives—such as completing a puzzle or assembling a component.

Motion Planning Engine
The trajectory generation component is responsible for planning collision-free and kinematically feasible paths in constrained environments. Algorithms such as RRT* (Rapidly-exploring Random Tree), A*, and optimization-based methods like CHOMP or STOMP are commonly used. The planner must account for manipulator constraints, task-specific precision requirements, and system dynamics.

Manipulation Controller
Precise object manipulation necessitates low-level control of the end-effector. This subsystem integrates feedback from tactile sensors (e.g., piezoelectric, capacitive) or visual markers (e.g., ArUco) to perform real-time adjustments. The goal is to ensure accurate placement and gentle handling of parts, avoiding damage or misalignment.

Human-Machine Interface (HMI)
To ensure practical deployment in semi-autonomous or supervised industrial environments, the system includes an HMI that allows users to set high-level goals, monitor progress, visualize perception outputs, and intervene when necessary. The interface may consist of a graphical user interface (GUI),

command console, or voice control, depending on the application requirements and user expertise.

5. Implementation Using Techman TM7S Cobot

5.1 Experimental Setup

The Techman TM7S collaborative robot offers a practical platform for implementing visuo-motor and cognitive assembly tasks. It combines a 6-DOF collaborative robotic arm with an integrated vision system and user-friendly programming environment.

For puzzle assembly by Techman Cobot TM7S [17] an experimental setup was configured as presented in the Fig. 4. The elements of the testbed system are (Fig. 4):

- Two high-resolution IDS uEye Cameras, Model UI-5240CP-P-HQ, captures the scene.
- Image processing identifies puzzle pieces, their edges, and potential neighbors.
- A deep learning classifier evaluates the most probable placement for each piece.
- The robot plans its movement to pick a selected piece and place it in the identified location using its vision-guided manipulation. The Fig. 5 presents a fragment of robotic puzzle dislocation from the *Staging Area* to the accurate position in the *Solution Area*. Robot controller receives the starting point coordinates A (x1, y1) as well as the end-point coordinates B (x2, y2) from the cameras (left and right one). The end-effector orientation (pneumatic gripper) should be kept always perpendicular to the robot work surface. The robot controller re-calculate coordinates of the points A and B in its' own coordinate system attached to the robot fundament.
- The Techman Cobot TM7S [17] controller does not support built-in path planning or end-effector trajectory generation. Therefore, this task is delegated to an external computer running MATLAB and the Robotics Toolbox for MATLAB/Simulink developed by Peter Corke [18]. The concept in this study involves using installed cameras to detect the location of a specific puzzle piece in the so-called *Staging Area* and the target position where the puzzle should be placed in the *Solution Area(Fig. 5)*. These coordinates are captured by two cameras, CAM-1 (left camera) and CAM-2 (right camera), and transmitted to the auxiliary computer, where they are used in MATLAB to plan the motion of the robot gripper within the task's operational workspace. Within MATLAB, the trajectory from point A to point B is computed based on inverse kinematics algorithms. In this experiment, artificial neural

networks were applied to calculate the internal joint coordinates (angles) of the robot, in order to accelerate the trajectory computation process. To achieve this, a neural network was first trained offline using a large dataset of robot positions within the task workspace, specifically where the end-effector was positioned in both the *Staging* and *Solution* areas for puzzle manipulation. Communication between MATLAB (on the auxiliary computer) and the Techman robot controller is established using the TCP/IP communication protocol.

- The robot performs the task of assembling the puzzle piece by piece in a sequential order, starting with the first piece that corresponds to the bottom-left corner of the final image.

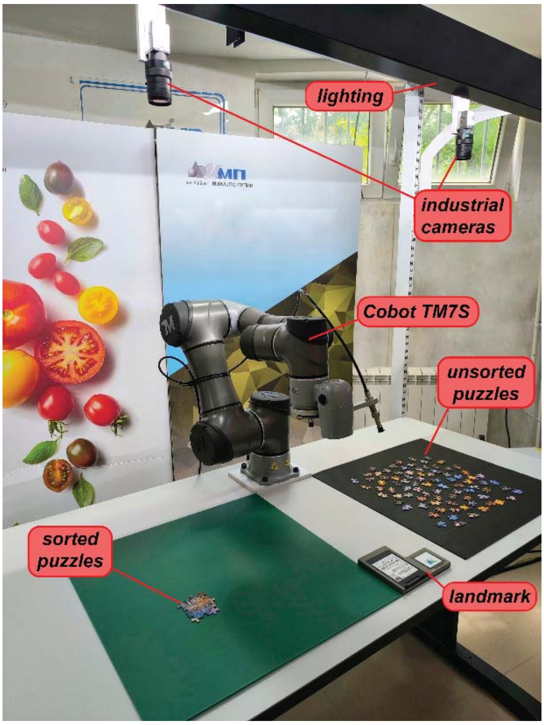


Figure 4. Experimental setup for robotic puzzle assembly with Cobot TM7S, cameras, lightining, landmarks and set of puzzles in the Stage Area (right) and Solution Area (left)

The procedure is repeated according to a predefined sequence: (i) perception, i.e., recognizing the next puzzle piece to be placed, following a left-to-right order, row by row, up to the top of the image; (ii) transmitting the coordinates (x1, y1) of the detected puzzle piece's centroid in the *Staging Area* and the target coordinates (x2, y2) in the *Solution Area* (Fig. 5), where the piece should be placed such that its centroid aligns precisely with this target point; (iii) based on these received coordinates, which represent the start and end points of the robot's trajectory, a nominal path is computed in MATLAB; (iv) the planned trajectory is then exported from MATLAB to the Techman robot controller, where it is executed; (v) the robot follows the calculated trajectory, while at the start point A, the pneumatic gripper receives a command to activate the vacuum and pick up the puzzle piece, and at the end point B, the gripper is commanded to release the vacuum, placing the piece accurately at the designated position.

5.2 Robot perception

The algorithm, implemented in MATLAB, relies on visual information to determine the placement of each puzzle piece in its predicted location(Fig. 4). The decision-making process is based on the following criteria:

1. the color of each segment/piece;
2. specific visual features present on the segments/pieces;
3. the shape of the puzzle pieces.

Assuming the puzzle consists of *n* individual pieces, the final (target) image is divided into *n* spatially and dimensionally aligned segments, such that each segment corresponds to exactly one puzzle piece and defines its intended location in the assembled image.

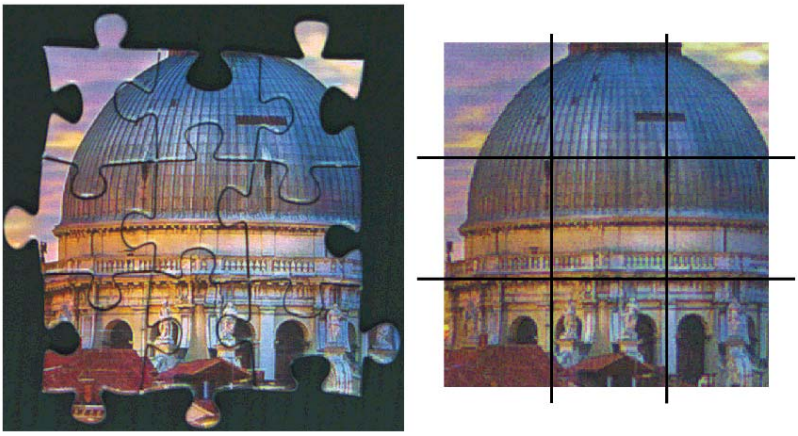


Figure 5. Example showing nine puzzle pieces (left) and their corresponding nine segments of the final (target) image (right)

The primary comparison criterion involves matching the red, green, and blue (RGB) color components between each segment and the corresponding puzzle piece. Another important criterion involves matching distinctive visual features or image patterns present on each segment and its corresponding puzzle piece.

However, these two criteria can often be disrupted by inconsistencies in shape—puzzle pieces are typically irregular, featuring interlocking tabs and slots. Therefore, shape compatibility becomes a crucial additional criterion.For instance, if k out of n puzzle pieces have already been assembled ($k < n$), the algorithm utilizes an overhead camera to scan the pieces and evaluate whether a new candidate piece fits spatially and geometrically as the $(k+1)^{th}$ piece in the puzzle.

5.3 Path Planing and Trajectory Generation

Trajectory planning for the Techman TM7S robot involves computing a smooth and feasible path for the robot's end-effector to move from a given start position to a target position in *Robot Operation Space*, both defined in Cartesian coordinates along with their respective orientations. The process begins by transforming these Cartesian poses into the robot’s joint space using inverse kinematics, which provides the joint configurations required at the start and end of the motion. A trajectory is then generated in joint space by interpolating between these configurations over time, ensuring continuous and smooth motion that adheres to the robot’s joint limits, velocity, and acceleration constraints. During this planning, collision avoidance with the robot’s own structure and surrounding environment is taken into account. The planned trajectory is typically optimized for efficiency, safety, and precision, and is executed through the robot’s control system, which ensures that each joint follows the calculated path accurately in real time.

When applying artificial neural networks (ANNs) to solve inverse kinematics for a robot like the Techman TM7S, the approach involves training a neural model to learn the complex, nonlinear relationship between the robot's end-effector pose (position and orientation in Cartesian space) and the corresponding joint angles. First, a large dataset is generated, typically using the robot’s forward kinematics equations, which compute the end-effector pose for known joint configurations. This data is used to train the ANN in a supervised learning setup, where the input to the network is the Cartesian pose, and the output is the corresponding set of joint angles. Once trained, the network can rapidly approximate joint configurations for any feasible pose within the robot’s workspace, bypassing the need for iterative numerical solvers. This method offers advantages in speed and adaptability, especially in real-time control scenarios or in applications where traditional inverse kinematics struggle due to redundancy or singularities. However, care must be taken to ensure the network generalizes well and that outputs remain within the robot's physical constraints.

5.4 Connecting Robot and Auxilary Computer

To connect a Techman TM7S robot with a computer running MATLAB, the method *TCP/IP socket communication*was used, the robot's control system had to be configured to accept incoming socket connections. This was done through the *TMflow interface* by adding a "Socket Listen" node with a specified port (e.g., 5890) and message format set to "String".

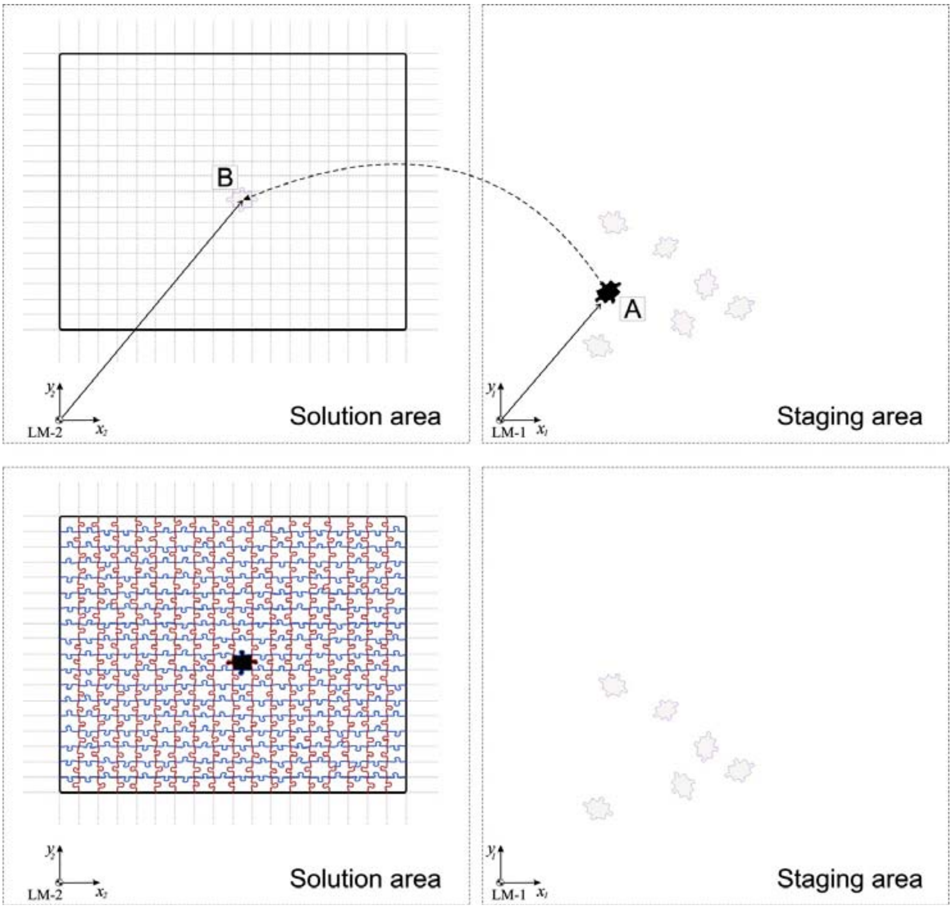


Figure 6. Visual-motor manipulation of puzzles from the Staging Area to the Solution Area.

The incoming messages were then passed to a "String to Command" node to interpret the received strings as robot commands (e.g., MoveJ, MoveL).
On the computer side, MATLAB used the *TCPclient function* to establish a connection to the robot's IP address and designated port. Commands generated in MATLAB (such as trajectory points from Peter Corke's Robotics Toolbox) were formatted as strings matching the robot's expected command syntax and sent over the TCP connection. This setup enabled MATLAB to stream joint or Cartesian commands in real time, allowing the TM7S robot to execute the desired trajectory, e.g. from point „A“ to point „B“ shown in Fig. 6.

Industrial analogs can be programmed using similar steps, where CAD data or visual templates replace the puzzle image.

6. Challanges and Considerations

Despite the growing capabilities of robotic systems, several critical challenges continue to impede the widespread and seamless deployment of intelligent robotic manipulators in complex environments. These challenges span across perception, decision-making under uncertainty, computational performance, and safety — especially in collaborative human-robot workspaces.

Perception in Cluttered Environments

One of the foremost challenges is achieving robust perception in real-world industrial settings that are often cluttered, dynamic, and partially observable. In puzzle-inspired robotic tasks, the robot must detect, recognize, and localize parts or objects in a scene filled with occlusions, overlapping components, reflective surfaces, and inconsistent lighting conditions. Traditional computer vision techniques often falter in such scenarios. Recent advances in deep learning and 3D vision, including the use of convolutional neural networks (CNNs), depth sensors, and attention mechanisms, have improved accuracy, but achieving consistent and reliable perception across varying contexts remains an open problem. Temporal coherence and sensor fusion approaches (combining visual, tactile, and even auditory data) are being actively explored to enhance robustness.

Uncertainty Handling

Industrial robots must operate under significant uncertainty due to sensor noise, variations in object placement, mechanical tolerances, and unpredictable external interactions. This uncertainty can lead to misclassifications, inaccurate localization, or failed grasps. Probabilistic models such as Bayesian networks, Monte Carlo localization, and partially observable Markov decision processes (POMDPs) offer theoretical tools to model and manage uncertainty. Moreover, data-driven approaches using reinforcement learning and imitation learning enable robots to adapt and improve performance over time, even under noisy conditions.

Real-Time Constraints
Many industrial applications impose stringent real-time requirements, where the robot must perceive, plan, and act within milliseconds. This necessitates highly optimized algorithms for perception, planning, and control, as well as efficient hardware integration. Balancing the trade-off between computational complexity and execution speed is essential. Edge computing, GPU-accelerated processing, and lightweight neural networks are emerging solutions that aim to deliver the necessary performance while maintaining power and resource efficiency. Additionally, techniques such as motion primitives and pre-computed action libraries are employed to speed up decision-making without sacrificing flexibility.

Safety and Compliance
As robots increasingly share physical spaces with humans, ensuring operational safety becomes paramount. Collaborative robots (cobots) must be equipped with capabilities for dynamic obstacle avoidance, compliant motion control, and human-intention prediction. Tactile sensors, force-torque sensors, and proximity detectors are commonly used to monitor interaction forces and adjust behavior in real time. Additionally, control strategies such as impedance control and admittance control allow the robot to respond flexibly to external disturbances, reducing the risk of injury or equipment damage. Regulatory standards such as ISO/TS 15066 provide guidelines for safety in human-robot collaboration and continue to shape the development of compliant robotic systems.

Addressing these challenges requires not only technical innovation but also interdisciplinary collaboration across robotics, artificial intelligence, human factors, and system integration. As robotic systems evolve, ongoing research must continue to push the boundaries of perception, decision-making, and interaction in order to achieve safe, adaptive, and intelligent behavior in complex industrial domains.

7. Experimental Results and Case Study

A simulated environment using the TM7S robot was constructed to perform puzzle assembly. Using a dataset of jigsaw piece images and a neural network trained for edge detection and piece matching, the robot successfully assembled puzzles with over 90% accuracy. The same framework can be adapted to a cable-routing task, showing successful generalization to an industrial scenario.

8. Future Work and Research Directions

As robotic systems continue to evolve, several key research directions emerge that aim to push the boundaries of robotic manipulation, autonomy, and adaptability in real-world industrial environments. These future developments are essential for enabling robots to handle increasingly complex tasks and collaborate more effectively with humans.

Multimodal Perception

One of the most promising avenues for enhancing robotic perception is the integration of multiple sensory modalities. While vision remains the primary source of information in many systems, combining it with tactile and auditory data can lead to a significantly richer and more robust understanding of the environment. For example, tactile sensors embedded in grippers can detect slippage, contact force, and material properties, enabling fine manipulation of delicate objects. Auditory cues, such as sounds generated during contact or motion, can also offer valuable information in determining success or failure in grasping tasks. Research in sensor fusion techniques, attention-based perception models, and cross-modal learning is critical to developing systems that can reason about the environment in a more human-like and adaptive manner.

Reinforcement Learning in Manipulation

Traditional model-based planning approaches often struggle to generalize across dynamic and unstructured environments. Reinforcement learning (RL), particularly deep reinforcement learning (DRL), has shown significant promise in enabling robots to learn optimal policies through trial-and-error interaction with their environment. In the context of robotic assembly or puzzle-like tasks, RL can be used to discover efficient action sequences, adapt to unknown object properties, and even recover from failures. Combining RL with imitation learning, hierarchical policy structures, and transfer learning techniques can further improve sample efficiency and generalization across tasks.

Scalability to Complex Assemblies

Current robotic assembly systems are typically limited to tasks involving a small number of components and fixed configurations. Future research must focus on scaling these systems to handle complex assemblies involving hundreds of parts, irregular geometries, and dynamically changing layouts. This requires improvements in task decomposition, modular planning, and memory-based reasoning. The development of scalable software architectures that support long-horizon planning, as well as efficient storage and retrieval of learned behaviors, is key to handling such complexity.

Human-Robot Collaboration: As manufacturing environments become increasingly flexible and decentralized, the role of collaborative robots (cobots) becomes more prominent. Future systems must be capable of intuitively understanding human intent, adapting to shared workspaces, and operating safely alongside human partners. This involves advancements in real-time human pose estimation, natural language understanding, shared autonomy, and adaptive behavior modeling. Effective human-robot collaboration also necessitates trust-building mechanisms, interpretable decision-making, and interactive learning, where robots can be guided and corrected by human workers in real time.

These research directions not only aim to improve technical performance but also aspire to create robotic systems that are socially and cognitively aware. The ultimate goal is to transition from task-specific automation to intelligent, general-purpose robotic co-workers capable of operating autonomously or cooperatively across a diverse range of industrial scenarios.

9. Conclusion

The puzzle assembly task offers a compelling metaphor for advanced robotic cognition and manipulation in industrial environments. Through the integration of AI, machine vision, and real-time planning, robots can transition from mere tools to intelligent collaborators. This methodology can be extended to a wide range of adaptive industrial applications, promoting efficiency, flexibility, and autonomy in modern manufacturing systems.

Aknowledgement

The results presented in this paper are results of the research supported by the Ministry of Science, Technological Development and Innovation of the Republic Serbia, under the contract no. 51-03-136/2025-03/200034from 04.02.205.This project was realized in cooperation with the company Renex within the mutual Agreement about R&D collaboration.

10. References

[1] Jigsaw Puzzle Robot. Last retirieved April 11th, 2025. https://github.com/JPStrydom/Jigsaw-Puzzle-Building-Robot

[2] CNC Jigsaw Puzzle Building Robot. Last retirieved April 11th, 2025. https://github.com/JPStrydom/Jigsaw-Puzzle-Building-Robot

[3] AI-Powered Jigsaw Puzzle Solving Robot. Last retirieved April 11th, 2025.

https://techmasterevent.com/project/ai-powered-jigsaw-puzzle-solving-robot

[4] P. Illig, R. Thompson, Q. Yu. Application of integral invariants to apictorial jigsaw puzzle assembly, Journal of Mathematical Imaging and Vision, 2023, Springer.

[5] Robotic Jigsaw Puzzle Solver Videos I. Last retirieved April 11th, 2025.https://www.youtube.com/watch?v=uDXAX4Dyg_4

[6] Robotic Jigsaw Puzzle Solver Videos II. Last retirieved April 11th, 2025. https://www.youtube.com/watch?v=gco7LGHw9Yg

[7] A. Zeng et al., "Learning Synergies Between Pushing and Grasping with Self-supervised Deep Reinforcement Learning," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4238–4245. DOI: 10.1109/IROS.2018.8594448

[8] M. Toussaint et al., "Differentiable physics and stable modes for tool-use and manipulation planning," *Robotics: Science and Systems (RSS)*, 2018. DOI: 10.15607/RSS.2018.XIV.057

[9] C. Finn, T. Yu, T. Zhang, P. Abbeel and S. Levine, "One-Shot Visual Imitation Learning via Meta-Learning," *Conference on Robot Learning (CoRL)*, 2017.

[10] J. Mahler et al., "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics," *Robotics: Science and Systems (RSS)*, 2017. DOI: 10.15607/RSS.2017.XIII.034

[11] D. Kragic and H. I. Christensen, "Survey on Visual Servoing for Manipulation," *Computational Vision and Active Perception Laboratory, CVAP/CAS*, 2002.

[12] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. DOI: 10.1038/nature16961

[13] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control," *arXiv preprint arXiv:1812.00568*, 2018.

[14] H. Van Hoof et al., "Learning Robot In-Hand Manipulation with Tactile Features," *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 121–127. DOI: 10.1109/HUMANOIDS.2015.7363540

[15] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal Deep Learning for Robust RGB-D Object Recognition," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 681–687. DOI: 10.1109/IROS.2015.7353447

[16] D. Katz, Y. Pyuro, and O. Brock, "Learning to Manipulate Articulated Objects in Unstructured Environments Using a Grounded Relational Representation," *Robotics: Science and Systems (RSS)*, 2008.

[17] Techman Cobot TM7S. Last retirieved April 25th, 2025. https://www.tm-robot.com/en/tm7s/

[18] Corke, P. I. (2017). *Robotics, Vision & Control: Fundamental Algorithms in MATLAB* (2nd ed.). Springer. ISBN: 978-3-319-54413-7.