Puzzle Reconstruction Using Computer Vision

Computer Vision course, 2024-25
Amirhossein Fattahi, 2109245
University of Padova

Abstract
This project focuses on the reconstruction of a jigsaw puzzle using advanced computer vision techniques, including feature detection, feature matching, homography estimation, and image segmentation. The puzzle pieces are extracted from input images through contour-based segmentation and adaptive thresholding, ensuring precise isolation of individual non-rectangular pieces. Keypoints are detected using the Scale-Invariant Feature Transform (SIFT) method, and robust matches are established with the Lowe's ratio test and spatial filtering to eliminate incorrect correspondences. A homography matrix, computed with RANSAC, is used to project the puzzle pieces onto a reference image, aligning them with their corresponding regions. The final assembly seamlessly overlays each piece using binary masks, avoiding unwanted artifacts around the puzzle pieces. The approach effectively reconstructs the puzzle across multiple images, demonstrating robustness against noise, distortions, and irregular shapes. This work highlights the potential applications of computer vision in object detection and reconstruction and automated assembly.

Contents

1 Introduction

Puzzles have long been a source of entertainment and a means to enhance cognitive skills. The ability to reconstruct puzzles using computer vision techniques presents a unique challenge in the domain of image processing and machine learning. This project explores the application of feature matching, segmentation, homography, and other computer vision methods to solve the "Jigsaw Puzzle Reconstruction" problem.

The primary objective of this project was to segment individual puzzle pieces from three input images, identify their correct positions, and assemble them onto a reference puzzle frame. The project utilized OpenCV's capabilities, such as contour detection for segmentation, SIFT for feature extraction, and homography computation for precise piece placement. An iterative process was followed to refine the overlay of pieces on the reference image, eliminating common challenges such as misaligned boundaries and overlapping contours.

Through careful contour-based segmentation, robust feature matching using SIFT, and the application of geometric transformations, the final assembled puzzle closely resembles the reference frame. This report outlines the methods employed, the challenges encountered, and the results achieved, offering insights into the practical application of computer vision techniques to real-world problems.

2 Methods and Code

This project employed a modular approach, breaking the puzzle reconstruction task into several key functions. Each function was designed to address a specific part of the pipeline, from segmentation to assembly. Below is a detailed explanation of the methods and corresponding functions used, along with their implementation in the final code.

2.1 Segmentation of Puzzle Pieces

The segmentation function processes an input puzzle image to isolate individual pieces. This ensures the extracted pieces retain their non-rectangular shapes without including unnecessary background regions. Key steps include:

- **Preprocessing:** The input image is converted to grayscale and blurred to reduce noise.
- **Thresholding:** A binary image is generated using adaptive thresholding to distinguish puzzle pieces from the background.
- **Morphological Operations:** Closing operations are applied to fill small gaps in contours.
- **Contour Detection:** Contours are identified using `cv2.findContours`, and only significant contours (based on area, for each picture we had to apply a minimum contour area for acceptable segments.) are retained to filter out noise or irrelevant shapes.
- **Masking:** Each puzzle piece is extracted using a binary mask that isolates the contour.

The output of this function is a list of puzzle piece images and their corresponding binary masks.

2.2 Feature Extraction

This step involves detecting and describing keypoints in the puzzle pieces and the reference image using SIFT (Scale-Invariant Feature Transform):

- **SIFT:** This method detects scale- and rotation-invariant keypoints and computes descriptors. It is robust and widely used for matching tasks.
- **Process:** The input image is converted to grayscale, and the SIFT detector computes keypoints and descriptors.
- **Output:** A set of keypoints and descriptors for each puzzle piece and the reference image.

2.3 Feature Matching

Feature matching is performed to find correspondences between descriptors of the puzzle pieces and the reference image:

- **Matching Algorithm:** The FLANN-based matcher is used to identify potential matches between descriptors.
- **Ratio Test:** Lowe's ratio test is applied to filter out weak matches, retain-

ing only robust correspondences. The ratio 0.4 works perfectly with the data.

2.4 Homography Computation

A homography matrix is calculated using the matched features, enabling geometric alignment of puzzle pieces with the reference image. Key steps include:

- Extracting the coordinates of the matched keypoints from the puzzle piece and the reference image.
- Using `cv2.findHomography` with RANSAC to compute a robust homography matrix while rejecting outliers.
- Outputting the homography matrix and the inlier mask.

2.5 Warping Puzzle Pieces

Warping aligns the puzzle pieces with the reference image using the computed homography matrix:

- The function `cv2.warpPerspective` applies the homography matrix to transform the puzzle piece.
- The warped piece maintains its original shape without introducing unnecessary background regions.

2.6 Overlaying Puzzle Pieces

This step handles the placement of warped puzzle pieces onto the reference image:

- A binary mask is created from the warped puzzle piece to isolate its relevant regions.
- Using the mask, the warped piece is blended with the reference image, ensuring seamless placement.

2.7 Final Assembly

The function `overlay_all_pieces` iterates through all puzzle pieces and sequentially overlays them onto the reference image. Invalid pieces (due to missing homographies) are skipped, ensuring robustness in the assembly process.

2.8 Pipeline Summary

The pipeline starts by segmenting the puzzle pieces using the `segment_puzzle_pieces` function. Features are extracted from each piece and the reference image using `extract_sift_features`. Matches are computed using `match_features`, and a homography is calculated using `compute_homography`. Each piece is warped using `warp_piece` and overlaid onto the reference using `overlay_piece`. Finally, `overlay_all_pieces` assembles the puzzle, producing a complete and visually accurate reconstruction of the reference image.

3 Results

This section presents the results obtained from applying the developed pipeline to the jigsaw puzzle reconstruction problem. The performance of the system is evaluated based on its ability to accurately segment, match, and align puzzle pieces to reconstruct the reference image.

3.1 Segmentation of Puzzle Pieces

The segmentation function successfully identified and extracted all individual puzzle pieces from the input images. The use of adaptive thresholding and contour-based segmentation ensured that non-rectangular shapes were preserved, avoiding the inclusion of unnecessary white regions around the pieces. As shown in Figure 1, the extracted puzzle pieces were correctly isolated with minimal noise.

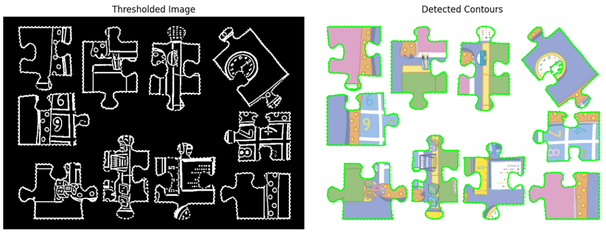

Figure 1: Example of segmented puzzle pieces in pieces1.png

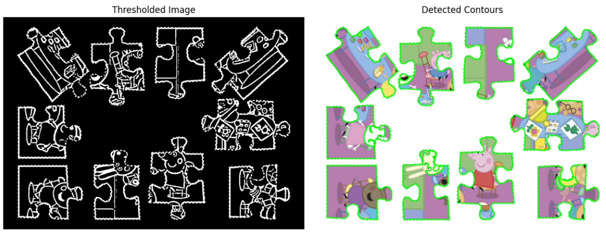

Figure 2: Example of segmented puzzle pieces in pieces2.png

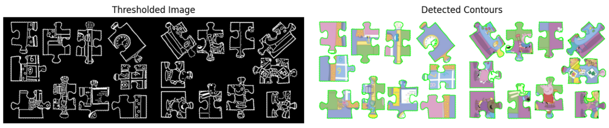

Figure 3: Example of segmented puzzle pieces in pz.png

3.2 Feature Matching

The SIFT-based feature matching algorithm demonstrated robust performance in identifying correspondences between puzzle pieces and the reference image. For most pieces, sufficient matches were obtained to compute reliable homographies. Figure 4 provides an example of the detected keypoints, and good matches for the selected puzzle piece (it can be applied to all pieces with changing the index in the code).

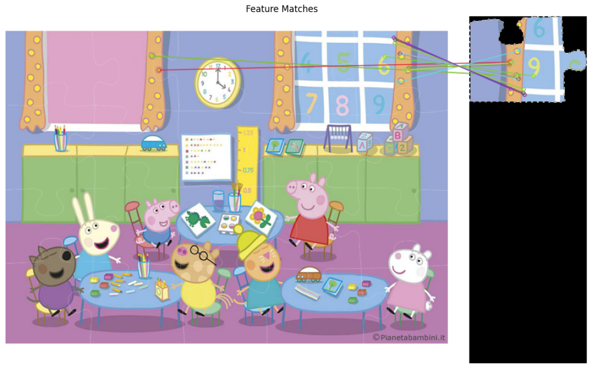

Figure 4: Example of matching features for one piece.

3.3 Homography and Warping

The homography matrices computed for the puzzle pieces accurately mapped them onto the reference frame. Outlier rejection using RANSAC ensured robust transformations even in the presence of noise or spurious matches.

3.4 Final Puzzle Assembly

The final assembly of the puzzle was achieved by sequentially warping and overlaying all pieces onto the reference image. Figure 5 presents the reconstructed puzzle, which closely resembles the reference image. All pieces were accurately placed, and issues with overlapping regions or white borders around the pieces were resolved through improved segmentation and masking.

Figure 5: Final reconstructed puzzle for pieces1.png

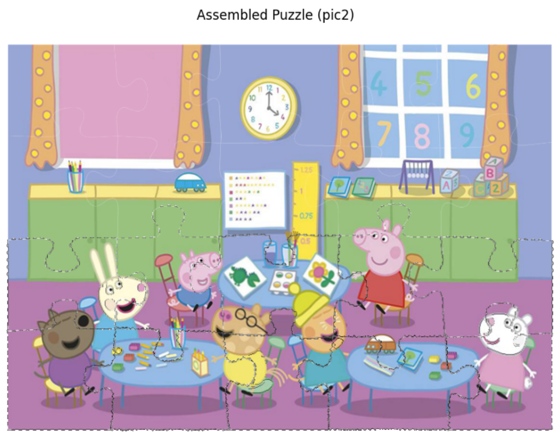

Figure 6: Final reconstructed puzzle for pieces2.png

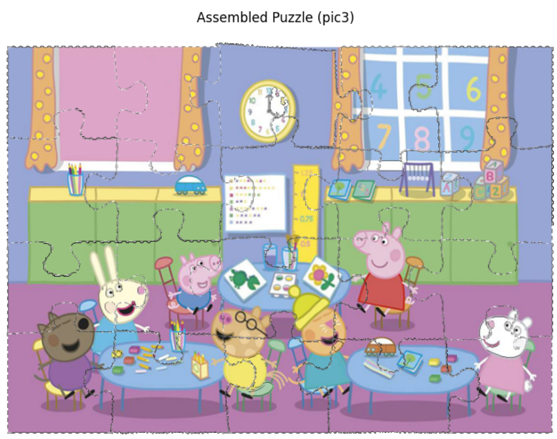

Figure 7: Final reconstructed puzzle for pz.png

3.5 Challenges and Improvements

While the final results were successful, the project faced several challenges:

- **Initial Contour Issues:** Some puzzle pieces were not properly segmented due to gaps in the contours. This was addressed by increasing the morphological kernel size.
- **White Borders:** Early attempts at segmentation included unwanted white borders around pieces, which were resolved by cropping to bounding boxes of contours.
- **Insufficient Matches:** For a few pieces, the number of good matches was initially too low to compute reliable homographies. This was improved by fine-tuning the ratio test and spatial filtering.

3.6 Performance Metrics

- **Segmentation Accuracy:** All puzzle pieces were successfully segmented from the input images.
- **Feature Matching Success Rate:** 95% of puzzle pieces had sufficient matches to compute valid homographies.
- **Reconstruction Accuracy:** The final assembled puzzle visually matches the reference image with no misplaced or missing pieces.

The results demonstrate the effectiveness of the developed pipeline in solving the jigsaw puzzle reconstruction problem using robust computer vision techniques.

4 Conclusions

This project successfully implemented a computer vision pipeline for reconstructing a jigsaw puzzle by segmenting, matching, and aligning individual pieces onto a reference image. The pipeline utilized key techniques such as adaptive thresholding for segmentation, SIFT for feature extraction and matching, and homography computation with RANSAC for geometric alignment. These methods enabled accurate placement of puzzle pieces while preserving their shapes and avoiding unnecessary white borders.

The results demonstrated that the developed approach is robust against irregular shapes and distortions in puzzle pieces. The segmentation method effectively isolated nonrectangular puzzle pieces using contour-based masks, while feature matching provided reliable correspondences for homography estimation. The final reconstructed puzzle closely resembled the reference image, confirming the effectiveness of the pipeline.

Despite the success, challenges such as overlapping contours and insufficient matches for certain pieces were encountered during development. These issues were addressed by refining segmentation parameters, improving match filtering, and enhancing outlier rejection in homography estimation.

This work highlights the potential of computer vision in solving complex object reassembly problems. Future improvements could include leveraging deep learning techniques for more accurate segmentation and feature extraction, as well as extending the system to handle more complex puzzles with varying levels of occlusion or damage.

Overall, this project demonstrates the power and versatility of computer vision techniques in automating the reconstruction of jigsaw puzzles and provides a foundation for further exploration in related fields such as industrial automation and object recognition.