**Similar Project Report**: <http://icvl.cs.bgu.ac.il/automatic-jigsaw-puzzle-solving/>

**Solving Jigsaw Puzzles Using Image Features** <http://www.sciencedirect.com/science/article/pii/S0167865508001931>

Method for automatic solving of jigsaw puzzles using image features instead of the shapes of the pieces

**Length:** 10 pages

**Reader:** Netta

**Notes:**

* Introduction
  + Humans - certainty of correctness (of two puzzle pieces fitting) vs. Computer - likelihood of correctness
  + Assumptions
    - Pieces have rigid edge curves constrained to a plane
    - Puzzle has no missing parts
    - Puzzle has the topology of a plane
  + Approach
    - Image features
      * Could improve the precision of the edge matching measure
      * Could be used in initial grouping of the pieces to reduce complexity when puzzle pieces are compared
    - Do not use the shape
    - Generalizes the problem so that puzzles with an edge description that is not available or not unique can be included (solved)
* Problem Definition
  + Assumption about the puzzle pieces
    - Each piece has four sides, all of which are arranged in a rectangular grid
    - Each side of a piece can be either concave, convex, or have a straight edge
    - A border piece has one or two straight edges
    - It is not required for the non-straight sides to be unique
  + Independent of Shape Uniqueness
    - Inclusion of puzzles that have perfectly rectangular pieces
      * Easy to generate (good for testing)
      * No distinctive edge, a lot harder to identify border pieces
* Related Work
  + Freeman and Gardner (1964)
    - Puzzles without pictures
    - Useful for segmentation
    - Calculating ow well pieces fit together - partial boundary curve matching
  + Radack and Badler (1982)
    - Partial boundary curve matching using polar coordinates
  + Wolfson et al. (1988)
    - Developed algorithm for solving puzzles using 2D Schwartz-Sharir curve matching & optimized search trees
  + Kosiba et al. (1994), Chung et al. (1998), Yao and Shao (2003)
    - Using image and shape of the pieces to correctly solve small puzzles
  + Goldberg et al. (2004)
    - Introduced method that used shape of the pieces and greedy approach (instead of a search tree)
* Solution Outline
  + Preprocessing - scanning real puzzles, separating the individual pieces, and rotating pieces to the proper alignment
  + Edge matching - matching an edge of one piece to an edge of another piece
  + Solution - ordering the pieces in such a way as to obtain the highest global match (complexity of O(n!))
  + Displaying the solution - includes minimizing overlaps and holes through corrections to the orientation of the pieces
* Info on the Test Data
  + Used two sets of puzzles for testing
  + First image was landscape and the second was of construction
* Edge Matching
  + Similarity Measure - how well two puzzle pieces match along a common edge
  + Used edge detector to determine if pieces fit together
    - Used filter based on the Sobel operator to get response vector
    - If two strips are not perfectly aligned, use a larger filter or can shift the strips relative to each other to find a local minimum response
    - Used response vector and mean square to get similarity measure in the range of [0,1]
  + Determining best image features to use for the similarity measure
    - Tested combinations of HSI (hue, saturation, and intensity) and RGB
  + For an edge - considered TP if the true match is also the edge with the highest similarity, otherwise it is considered FN
  + Used TPR to indicate reliability
* Edge Sampling
  + Possible Issues
    - Scanned pieces can have leftover cardboard fragments, which can lead the digital segmentation to have artifacts along the edges of pieces
    - Imprecise segmentation can cause problems near the edges of some pieces
    - Some images of pieces may include the side of the pieces due to an oblique angle with the camera
  + Method
    - Step 1 - determines location of the “corners” of the piece
      * Limit the search to a sub-image
    - Step 2 - use corner points as start and end points of the edge
      * Compute points along the edge so that they are placed exactly on the contour of each piece on the outer edge of every pixel (ensures same # of samples for perfectly matching concave & convex edges)
    - Step 3 - estimating a tangent from each of the points along the contour
      * Uses secant spanned by points on either side
      * Distance between the points depends on the size of the puzzle piece
    - Step 4 - determining value of edge strip
      * Calculated by linear interpolation using the neighboring pixels of the points perpendicular to the tangent
    - Non-matching edges likely to generate edge strips of different lengths, but matching edges can also differ
      * Solved by centering two strips relative to each other and truncating longer strip by an equal amount at each end
* Edge Matching Results
  + Edge matching
    - Compute similarity measure that indicates how well two pieces fit together
    - Value of similarity measure should be close to 1 for the matching edges
  + Performs better on computer generated puzzles than on real puzzles
* Solving Puzzles
  + Singlepiece Algorithm
    - Step 1 - solve the puzzle border using the assignment problem heuristic
      * If more than one “tour” is found then merge them by testing all of the valid combinations
    - Step 2 - repeats until all of the pieces have been placed
      * Part A - find all pockets (empty positions with two adjacent pieces)
      * Part B - for each pocket, place the best matching piece using the matching algorithm and similarity function
* Puzzle Solver Results
  + Success was measured by counting the incorrectly placed pieces when the algorithm stops
* Real Puzzles
  + Problems with perspective
    - Minimized by digitizing one piece at a time and at a fixed distance
  + Problem with brightness
    - Best way to avoid is by disabling automatic brightness
    - Can also be avoided if only the hue color channel is used
  + Other problems
    - Use of Gaussian smoothing filter to compensate for edge artifacts

**An Innovative Algorithm for Solving Jigsaw Puzzles Using Geometrical and Color Features**

<https://link.springer.com/content/pdf/10.1007%2F11578079_99.pdf>

The whole method concerns a recurrent algorithm, which initially, finds the most important corner points around the contour of a piece, afterwards performs color segmentation

**Length:** 11 pages

**Reader:** Eric

**Notes:**

**Automatic Jigsaw Puzzle Solver**

<https://nithyanandabhat.weebly.com/uploads/4/5/6/1/45617813/project_report-jigsaw-puzzle.pdf>

This is a student’s final project, solves small (12 piece) puzzles from a photo of the pieces. Could be used to generate ideas for the structure of our algorithm.

**Length:** 23 pages

**Reader:** Ethan

**Notes:**

**Summary** This paper is about a puzzle solver using image qualities and shape, but operates under the assumption that the puzzle is only 12 pieces. This discusses their process for edge matching: creating a mask, extracting edge profile data, determining corners, rotating pieces, comparing with other pieces for matches. They generate a shorter list of pieces for which the edges match. They then compare to a reference image to see which piece is the correct match. They discuss a possible way to match the image using a contour plot along the boundary region.

**Applicability** This is basically a very small scale version of what we are trying to accomplish. Having an outline of a comparable algorithm will be useful for structuring our algorithm. Their work on shape characteristics will be a useful starting point for our shape analysis.

**Issues** They make many assumptions that we would rather not make, like limiting the number of pieces to 12. Also, they do not implement the comparison of the colors of edges without a reference image. This is somewhat poorly written and I would not trust it very much, other than a vague idea of outlining the process.

**Related Work**

[2a] www.math.umn.edu Article Title: Automatic Solution of Jigsaw Puzzles Retrieved from http://www.math.umn.edu/~olver/vi\_/puzzles.pdf

**Jigsaw Puzzles with Pieces of Unknown Orientation**

<http://chenlab.ece.cornell.edu/people/Andy/Andy_files/Gallagher_cvpr2012_puzzleAssembly.pdf>

**Length:** 8 pages

**Reader:** Nobody

**Notes:**

**Solving Square Jigsaw Puzzles with Loop Constraints**

<https://www.cc.gatech.edu/~hays/papers/puzzle_eccv14.pdf>

**Length:** 15 pages

**Reader:** Scott

**Notes:**

**Summary:** Ummmmm

**Applicability:** Ummmmm

**Issues:** Ummmmmm