Clustering-Based, Fully Automated Mixed-Bag Jigsaw Puzzle Solving

Zayd Hammoudeh   Chris Pollett

Department of Computer Science
San José State University
San José, CA USA

17th International Conference on Computer Analysis of Images and Patterns
August 22-24, 2017

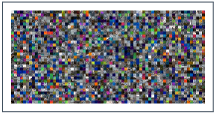
- First jigsaw puzzle introduced in the 1760s
- First computational jigsaw puzzle solver introduced in 1964 [4]
- Solving a jigsaw puzzle is NP-complete [1, 3].
- **Example Applications:** DNA fragment reassembly, shredded document reconstruction, and speech descrambling
  - Generally, the ground-truth source is unknown.

Jig Swap Puzzles: Variant of the traditional jigsaw puzzle

- All pieces are equal-sized squares.
- **Piece rotation, puzzle dimensions, and ground-truth input contents are all unknown.**

"**Mixed-Bag**": Simultaneous solving of **multiple** jig swap puzzles

- The number of inputs may be unknown.



Randomized Solver Input – 2,017 Pieces

Solver Output #1
805 Pieces

Solver Output #2
540 Pieces

Solver Output #3
672 Pieces

Summary of Key Contributions

- **Primary Contribution:** Novel mixed-bag puzzle solver that outperforms the current state of the art [6] by:
  - Requiring no external "oracle" information
  - Generating superior reconstructed outputs
  - Supporting more simultaneous inputs

- **Additional Contribution**: Define the first metrics that quantify the quality of outputs from a multi-puzzle solver

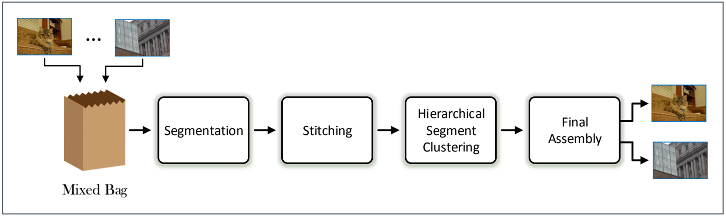
Our Contribution:

The Mixed-Bag Solver

Basis of the Mixed-Bag Solver: Human puzzle solving strategy to:

- Correctly assemble small puzzle regions (i.e., segments)
- Iteratively merge smaller regions to form larger ones

Simplified Algorithm Flow:



Segmentation

Mixed-Bag Solver Stage #1

- **Segment**: Partial puzzle assembly where this is a high degree of confidence pieces are placed correctly
  - Each piece is assigned to at most one segment.
- **Role of Segmentation**: Provide structure to the set of puzzle pieces by partitioning them into disjoint segments

Segmentation
Algorithm Overview

- Iterative process consisting of one or more rounds.
- In each round, any pieces not already assigned to a segment pieces are assembled into a single puzzle.
  - This assembly is then segmented based on inter-piece similarity (i.e., the "best buddies" principle).
  - Segments of sufficient size are saved for use in later Mixed-Bag Solver stages.
- Segmentation terminates when an assembly has no segments whose size exceeds a minimum threshold (e.g., 7).

Clustering-Based, Fully Automated Mixed-Bag Jigsaw Puzzle Solving

Hammoudeh & Pollett

Introduction

Mixed-Bag Solver

Segmentation

Stitching

Hierarchical Clustering

Quantifying Quality

Direct Accuracy

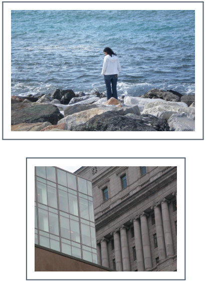
Experimental Results

Input Puzzle Count

Solver Comparison

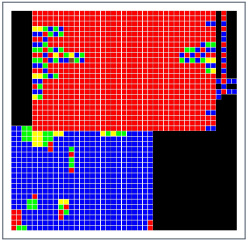
References

7



Ground-Truth Inputs

Solver Output



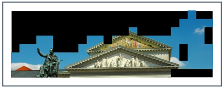
Segmented Output

- **Role of Stitching**: Quantify the extent that any pair of segments is related
- **Mini-Assembly**: Places a pre-defined, fixed number (e.g., 100) of pieces
- **Stitching Piece**: A piece near the boundary of a segment that is used as the seed of a single mini-assembly
- **Segment Overlap**: Inter-segment affinity score based on the composition of a segment's mini-assembly

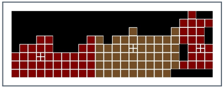
Stitching
Example – Single Input Image

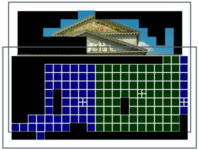
Ground Truth

Segmenter Output



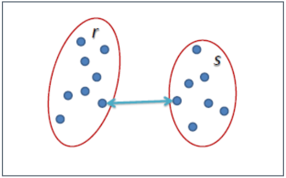
Stitching Pieces Mini-Assembly





Stitching piece selected from upper-right corner of the top segment

- A single ground-truth image may be comprised of multiple segments.
- **Role of Hierarchical Clustering**: Estimate the number of inputs by grouping together all segments from the same ground-truth image.
- **Single-Link Clustering**: Inter-cluster similarity equals the similarity of their most similar respective members



- The solver continues merging segment clusters until one of two criteria is satisfied:
  - Only a single segment cluster remains
  - Maximum similarity between any segment clusters is below a predefined threshold
- **Final Assembly**: Builds the final solver outputs are built using the cluster membership results

Quantifying Solver Performance

Quantifying Solver Performance

- Metrics quantify the quality of the solver outputs as the reconstructions may not be reconstructions.
- **Two Primary Quality Metrics**: Range [0,1]
  - Direct Accuracy
  - Neighbor Accuracy (*not discussed in this presentation*)
- **Disadvantages of Current Metrics**: Neither account for issues unique to mixed-bag puzzles including:
  - Pieces from one input misplaced in multiple output puzzles
  - Pieces from multiple inputs in the same output

Direct Accuracy

Overview of the Current Standard

Standard Direct Accuracy: Fraction of pieces, c placed in the same location in both the ground-truth and solved puzzles versus the total number of pieces, n

Formal Definition:

$$DA = \frac{c}{n} \tag{1}$$

SEDAS: A new quality metric with two primary improvements over standard direct accuracy:

$$SEDAS_{P_i} = \max_{l \in L} \left(\max_{S_j \in S} \frac{c_{i,j,l}}{n_i + \sum_{k \neq i}(m_{k,j})} \right) \quad (2)$$

- **Mixed-Bag Support**: For input, $P_i \in P$, and output, $S_j \in S$, penalize for missing pieces (via n_i) and additional pieces (via $\sum_{k \neq i} m_{k,j}$)
- **Shiftable Reference**: Shift the direct accuracy reference coordinate, l within a set of possible puzzle piece locations, L, ($l \in L$), in order to maximize the overall score

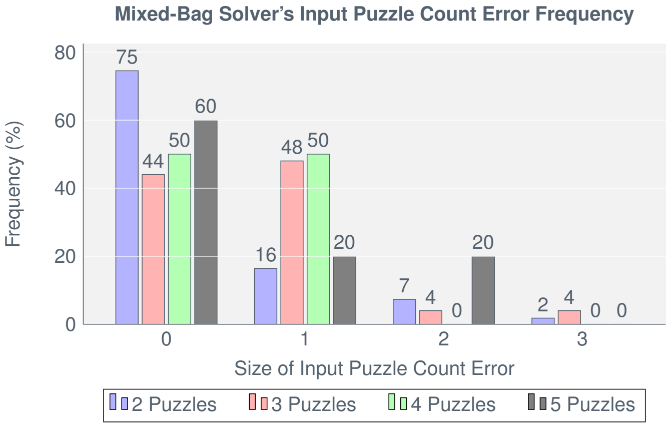
Experimental Results

Overview of the Experiments

- **Standard Jig Swap Puzzle Experiment Conditions**: Defined by Cho *et al.* (CVPR 2010) [2] and followed by [7, 6, 9, 5]
- **Procedure**: Randomly select, without replacement, a specified number of images (between 2 and 5) from the 805 piece, 20 image data set [8]
- **Two Primary Experiments:**
  - Estimation of the Ground-Truth Input Count
  - Comparison of Overall Reconstruction Quality
    - **Baseline**: Current State of the Art - Paikin & Tal (CVPR '15) [6]
    - **Our Competitive Disdvantage**: Paikin & Tal's algorithm had to be provided the number of input puzzles.

Estimating the Ground-Truth Input Count

Multiple Input Puzzles



Puzzle Count Error: Difference between the actual number of input puzzles and the Mixed-Bag Solver's estimate

Overall Accuracy: 65%

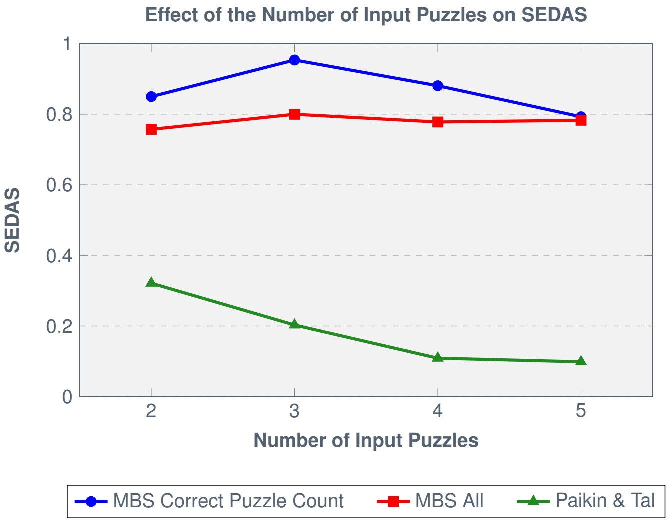
Comparison of Reconstruction Quality

Performance on Multiple Inputs

- **Goal**: Compare the quality of the outputs from the Mixed-Bag Solver (**MBS**) and Paikin & Tal's algorithm
- **Note**: Our Mixed-Bag Solver's performance when it correctly estimated the puzzle count is also shown.
  - This is an approximate representation of the performance had there been optimal hierarchical clustering.

Comparison of Reconstruction Quality

Shiftable Enhanced Direct Accuracy Score (SEDAS)



Performance on Multiple Input Puzzles

Results Summary

- **Summary**: Our Mixed-Bag Solver significantly outperforms the state of the art, Paikin & Tal.
  - This is despite their algorithm having a competitive advantage by being supplied the number of input puzzles.
- **Puzzle Input Count**: Our approach shows no significant performance decrease with additional input puzzles.
- **Effect of Clustering Errors**: Performance only decreased slightly when incorrectly estimating the input puzzle count
  - Many of the extra puzzles were relatively insignificant in size.

List of References I

[1] Tom Altman. Solving the jigsaw puzzle problem in linear time. *Applied Artificial Intelligence*, 3(4):453–462, January 1990. ISSN 0883-9514.

[2] Taeg Sang Cho, Shai Avidan, and William T. Freeman. A probabilistic image jigsaw puzzle solver. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '10, pages 183–190. IEEE Computer Society, 2010.

[3] Erik D. Demaine and Martin L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23 (Supplement):195–208, June 2007.

[4] H. Freeman and L. Garder. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers*, 13:118–127, 1964.

[5] Andrew C. Gallagher. Jigsaw puzzles with pieces of unknown orientation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 382–389. IEEE Computer Society, 2012.

[6] Genady Paikin and Ayellet Tal. Solving multiple square jigsaw puzzles with missing pieces. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '15. IEEE Computer Society, 2015.

[7] Dolev Pomeranz, Michal Shemesh, and Ohad Ben-Shahar. A fully automated greedy square jigsaw puzzle solver. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '11, pages 9–16. IEEE Computer Society, 2011.

[8] Dolev Pomeranz, Michal Shemesh, and Ohad Ben-Shahar. Computational jigsaw puzzle solving. `https://www.cs.bgu.ac.il/~icvl/icvl_projects/automatic-jigsaw-puzzle-solving/`, 2011. (Accessed on 05/01/2016).

[9] Dror Sholomon, Omid David, and Nathan S. Netanyahu. A genetic algorithm-based solver for very large jigsaw puzzles. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '13, pages 1767–1774. IEEE Computer Society, 2013.