

An Enhanced Jigsaw Puzzle Solver

CS297 Final Report

Zayd Hammoudeh
(zayd.hammoudeh@sjsu.edu)

May 1, 2016

Contents

List of Figures	ii
1 Introduction	1
2 Previous Work	2
3 Puzzle Piece Pairwise Affinity	4
3.1 Cho <i>et. al.</i> Pairwise Affinity	4
3.2 Pomeranz <i>et. al.</i> Pairwise Affinity	4
3.3 Paikin & Tal Pairwise Affinity	5
3.3.1 Improved Asymmetric Compatibility	6
3.4 Best Buddies	7
3.4.1 Visualizing Best Buddies	7
3.4.2 Unique Best Buddies	8
3.4.3 Interior and Exterior Best Buddies	9
4 Quantifying Solver Accuracy	10
4.1 Direct Accuracy	10
4.1.1 Enhanced Direct Accuracy Score	10
4.1.2 Shiftable Enhanced Direct Accuracy Score	11

List of Figures

1	A Computer Manipulated Image with Poor Asymmetric Compatibility	6
2	Visualization of Best Buddies in an Image	8
3	Solver Output where a Single Misplaced Piece Catastrophically Affects the Direct Accuracy	12

1 Introduction

Jigsaw puzzles have been around since the 1760s when they were made from wood. Their name derives from the fact that they were originally carved using jigsaws. The 1930s saw the introduction of the modern jigsaw puzzle where an image was printed on a cardboard sheet that was cut into a set of interlocking pieces [1, 2]. Although jigsaw puzzles had been solved by children for centuries, it was not until 1964 that the first automated jigsaw puzzle solver was proposed by [3], and that solver could only solve 9 piece puzzles. While an automated jigsaw puzzle solver may seem trivial, it has been shown by [4] and [5] to be strongly NP-complete when pairwise compatibility between pieces is not a reliable metric for determining adjacency.

A jig swap puzzles are specific type of jigsaw puzzle where all pieces are equally sized, non-overlapping squares. Jig swap puzzles are substantially more difficult to solve than standard jigsaw puzzle as one can not consider mechanical compatibility when trying to determine affinity between pieces. As such, one can only consider the image information on each individual piece when solving the puzzle.

Solving a jigsaw puzzle simplifies to reconstructing an object from a set of component pieces. As such, techniques developed for jigsaw puzzles can be generalized to many practical problems. Examples where jigsaw puzzle solving strategies are applicable include: reassembly of archaeological artifacts [6, 7], forensic analysis of deleted files [8], image editing [9], reconstruction of shredded documents [10], DNA fragment reassembly [11], and speech descrambling [12].

Unlike traditional jigsaw or jig swap puzzles, the ground-truth (i.e. original) image is unknown for most practical applications. This significantly complicates solving the problem as one must determine the overall structure of the complete solution solely from a bag of individual pieces with unknown relationships.

This project proposes an improved jig swap puzzle solver. It also studies the fundamental structures within images to determine which images are easy or difficult to solve using current techniques. This study will help guide future research by focusing on the weaknesses with state of the art approaches.

2 Previous Work

Computational solvers for jigsaw puzzles have been studied since the 1960s when Freeman and Garder proposed an approach that could solve jigsaw puzzles of up-to nine pieces using solely the piece shapes [3]. Since then the focus of research has shifted from traditional jigsaw puzzles to jig swap puzzles.

Cho *et. al.* [13] proposed in 2010 one of the first modern computational jig swap puzzle solvers; their approach relied on a graphical model build around a set of one or more “anchor piece(s)”; these one or more anchor pieces were fixed in their correct location before the solver began. Cho *et. al.* also assumed knowledge regarding the size of the puzzle dimensions. Future solvers would on Cho *et. al.*’s results while also reducing the amount of information passed to the solver beyond the set of pieces.

A significant contribution of Cho *et. al.* is that they were first to use the LAB (L_{ightness} and the A/B opponent color dimensions) colorspace to encode image pixels (as opposed to standards such as RGB or CMYK) since LAB normalizes the lightness and color variation across all three dimensions. Cho *et. al.* proposed a measure for quantifying the pairwise distance between two puzzle pieces that became the basis of most of the future work (see 3 for more details).

Pomeranz *et. al.* [14] proposed an iterative, greedy jig swap puzzle solver in 2011. Their solver did not rely on anchor pieces, and the only information passed to the solver were the pieces and the size of the puzzle. Pomeranz *et. al.* also generalized and improved on Cho *et. al.* piece pairwise distance measure by proposing a “predictive distance measure”. Finally, Pomeranz *et. al.* proposed the concept of “best buddies”, which are two pieces that are more similar to each other than they are to any other pieces. Best buddies have served as the basis for both an estimation metric for the quality of a solved image as well as the foundation of some solvers’ placers [15].

In 2012, Gallagher [16] formalizes the different categories of jig swap problems into three primary types. The following is Gallagher’s proposed terminology, and his nomenclature is used throughout this document.

- **Type 1 Puzzle:** The dimension of the puzzle (i.e. the width and height of the ground-truth image in number of pixels) is known. What is more, the orientation of each piece is known, which means that there are exactly four pairwise relationships between any two pieces. A single anchor piece, with a known, correct, location is required with additional anchor pieces being optional. This type of puzzle is used by [13, 14].
- **Type 2 Puzzle:** This is an extension of a type 1, where pieces may be

rotated in 90° increments (e.g. 0° , 90° , 180° , or 270°). This change alone increases the number of possible solutions by a factor of 4^n (where n is the number of pieces in the puzzles) in comparison to a Type 1 puzzle. What is more, no piece locations are known in advance; this change eliminates the use of anchor piece(s). Lastly, the dimensions of the ground-truth image/puzzle can be unknown.

- **Type 3 Puzzle:** All puzzle piece locations are known and only the rotation of the puzzle pieces is unknown. This is the least computationally complex of the three puzzle variants and is generally considered the least interesting. Type 3 puzzles are not explored as part of this thesis.

Sholomon *et. al.* also propose a fourth type (i.e. "Type 4") puzzle whereby each puzzle piece has an image on both sides. Each puzzle is similar to a

3 Puzzle Piece Pairwise Affinity

Pairwise affinity quantifies the similarity between the edges of two puzzle pieces. $D(x_i, s_i, x_j, s_j)$ and $C(x_i, s_i, x_j, s_j)$ represent the distance and compatibility (i.e. similarity) between side s_i of puzzle piece x_i and side s_j of puzzle piece x_j .

3.1 Cho *et. al.* Pairwise Affinity

As mentioned in section 2, Cho *et. al.* [13] proposed one of earliest edge-based pairwise affinity measures for two puzzle pieces. Their approach is shown in Equation (1), which quantifies the distance between the left (“ L ”) side of piece x_i and the right (“ R ”) side of piece x_j . K is the width/height of a puzzle piece in number of pixels¹.

$$D(x_i, L, x_j, R) = \sum_{k=1}^K \sum_{d=1}^3 (x_i(k, K, d) - x_j(k, 1, d))^2 \quad (1)$$

Since the LAB colorspace has three dimensions (e.g. lightness and A/B opponent colors), d ranges between 1 and 3. Similarly, $x_i(k, K, d)$ represents the LAB value in dimension d of the pixel in row “ k ” of column K in piece x_i .

3.2 Pomeranz *et. al.* Pairwise Affinity

One of the disadvantages of Cho *et. al.*’s metric is that it simply squares the difference between the piece’s pixel dimensions. It is possible that superior results may be achieved by allowing the solver to modify this exponent term. Pomeranz *et. al.* in [14] generalize Equation (1) using the L_p norm as shown in Equation (2)².

$$D(x_i, L, x_j, R) = \left(\sum_{k=1}^K \sum_{d=1}^3 (|x_i(k, K, d) - x_j(k, 1, d)|)^p \right)^{\frac{q}{p}} \quad (2)$$

Hence, for a distance measure equivalent to that of Cho *et. al.*, p and q are set equal to 2.

¹Cho *et. al.* used 7 for K .

²Pomeranz *et. al.* used 28 for K . This has generally become the standard in subsequent papers.

An additional down side of the metric proposed by Cho *et. al.* is that it only considers the border pixels. Hence, if there is some gradient in the ground-truth image, two pieces may appear artificially dissimilar using this approach. To address this issue, Pomeranz *et. al.* proposed predictive compatibility. Equation (3) shows the predictive compatibility between the left (“*L*”) side of piece x_i and the right (“*R*”) side of piece x_j .

$$C(x_i, L, x_j, R) = \sum_{k=1}^K \sum_{d=1}^3 \left[([2x_i(k, K, d) - x_i(k, K - 1, d)] - x_j(k, 1, d))^p - ([2x_j(k, 1, d) - x_i(k, 2, d)] - x_i(k, K, d))^p \right]^{\frac{q}{p}} \quad (3)$$

Note that in Equation (3), the difference between the column of pixels adjacent to the edge is used to account for any gradient effects.

3.3 Paikin & Tal Pairwise Affinity

Paikin & Tal in [15] used Pomeranz *et. al.*’s predictive compatibility as the foundation of their asymmetric distance measure. Paikin & Tal’s approach is shown in Equation (4).

$$D(x_i, L, x_j, R) = \sum_{k=1}^K \sum_{d=1}^3 \| [2x_i(k, K, d) - x_i(k, K - 1, d)] - x_j(k, 1, d) \| \quad (4)$$

Note that Paikin & Tal set p and q equal to 1 as it not only increased the accuracy of the solver but also significantly reduced the computational time. It is important to note that since this distance is asymmetric, in most cases $D(x_i, L, x_j, R)$ will not equal $D(x_j, R, x_i, L)$.

Pomeranz *et. al.* only considered the relationship between two individual pieces when determining the pairwise compatibility. In images (or areas of images) with little variation (e.g. an all white image), parts may have artificially high pairwise distances. To account for this, Paikin & Tal proposed asymmetric compatibility as shown in Equation (5).

$$C(x_i, L, x_j, R) = 1 - \frac{D(x_i, L, x_j, R)}{\text{second}D(x_i, L)} \quad (5)$$



Figure 1: A Computer Manipulated Image with Poor Asymmetric Compatibility

where $secondD(x_i, L)$ is the second best asymmetric distance between the left side of piece x_i and all other pieces. By normalizing compatibility with respect to the second best match, it is possible to identify those pairings that have actually have high compatibility while at the same time removing the artificially high compatibilities between pieces from areas of the image with low variation.

3.3.1 Improved Asymmetric Compatibility

In an image that was generated from an analog input (e.g. a photograph), it is expected that there will be some degree of natural variation across the image due to variations in brightness, the object in the photo, and the image sensor. However, in digital images that are generated or manipulated by computers, this variation can be trivially removed. An example of this would be an image of a solid color or object(s) in front of a white background as shown in figure 1.

The puzzles pieces along the border of the image as well as the pieces in the background have sides that are all white. This entails that regardless of which of the three metrics are used, the distance between them is zero. What is more, Paikin & Tal do not explicitly define how to handle this case for asymmetric compatibility. Therefore, to better handle computer generated images, this thesis proposes an enhanced definition of asymmetric compatibility as shown in Equation (6).

$$C(x_i, L, x_j, R) = \begin{cases} 1 - \frac{D(x_i, L, x_j, R)}{secondD(x_i, L)} & secondD(x_i, L) \neq 0 \\ -\alpha & secondD(x_i, L) = 0 \end{cases} \quad (6)$$

where α is a penalty term that denotes that such pairings have low compatibility.

3.4 Best Buddies

As defined by Pomeranz *et. al.*, two pieces, x_i and x_j are best buddies on their respective sides s_i and s_j if they are more compatible (i.e. similar) to each other than they are to each other. This is more formally described in Equation (7). This definition of best buddies applies regardless of whether the compatibility definition of Pomeranz *et. al.* or Paikin & Tal is used.

$$\begin{aligned} \forall x_k \forall s_k \in Parts, \quad C(x_i, s_i, x_j, s_j) &\geq C(x_i, s_i, x_k, s_k) \\ \text{and} \\ \forall x_k \forall s_k \in Parts, \quad C(x_j, s_j, x_i, s_i) &\geq C(x_j, s_j, x_k, s_k) \end{aligned} \tag{7}$$

Parts represents the set of all pieces in the puzzle, and s_k is any of the four sides of piece x_k .

In most cases, it is relatively rare that two pieces are best buddies and are not actually neighbors [15]. When considering all sides of a piece, it is rarer still that a piece has more best buddies that are not its neighbor than that are its neighbor. This makes best buddies a critical tool in many placers.

3.4.1 Visualizing Best Buddies

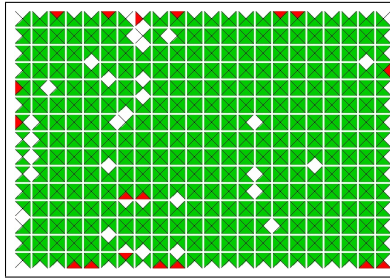
There is currently no standard for visualizing best buddy relationships in images. This thesis proposes a format for the first time. As a nomenclature, this thesis refers to any best buddies that are neighbors in the image as “adjacent best buddies” while any best buddies that are not neighbors are referred to as “non-adjacent best buddies”.

In a jig swap puzzle, a piece may have best buddies on up to four sides (since the pieces are square). As such, each piece in the best buddy visualization is divided into four isosceles triangles; the base of each triangle is along the side of the puzzle piece whose best buddy is being denoted as either adjacent or non-adjacent. The four isosceles triangles all share a common, non-base vertex in the center of the puzzle piece. A white border is placed around all puzzle pieces to make it easier to differentiate pieces from one another. A black square represents a missing or unpopulated piece location.

Figure 2 shows an image denoted as “(a)” and its accompanying best buddy visualization denoted as “(b)”. Adjacent and non-adjacent best buddies are



(a) Original Image



(b) Best Buddy Visualization

Figure 2: Visualization of Best Buddies in an Image

represented by green and red isosceles triangles. Any edge that has no best buddies is shown as a white triangle.

3.4.2 Unique Best Buddies

As explained in Section 3.3.1, reconstructing computer generated or computer manipulated image may introduce new challenges not found with photographs; best buddies is another example of this.

In photographs, it is generally unlikely that there will be a significant number of cases where a piece has multiple best buddies on the same side. However, computer generated images may have areas of truly solid colors, and any pieces in those areas will all be best buddies with one another, making best buddies a far less discerning indicator of compatibility. This thesis addresses such issues by modifying the definition of best buddies as shown in Equation (8).

$$\begin{aligned}
&\forall x_k \forall s_k \in Parts, \quad C(x_i, s_i, x_j, s_j) > C(x_i, s_i, x_k, s_k) \\
&\text{and} \\
&\forall x_k \forall s_k \in Parts, \quad C(x_j, s_j, x_i, s_i) > C(x_j, s_j, x_k, s_k)
\end{aligned} \tag{8}$$

Rather than relying on best buddies being “greater than or equal” to all other pieces, the modified requirement is that pairings must be strictly greater than. Hence, best buddy pairings are exclusive with cliques of size greater than two being explicitly disallowed.

3.4.3 Interior and Exterior Best Buddies

In all previous research, all best buddies (in particular best buddy errors) were treated the same. However, pieces that are on the exterior (i.e. edge) of a puzzle or that are next to a missing piece are expected to naturally have a higher rate of non-adjacent best buddies than pieces on the interior of the puzzle. This is because such piece sides do not have a true neighbor leaving them more likely to couple with an unrelated piece. As an example, Figure 2 has four non-adjacent interior best buddies and 14 exterior non-adjacent best buddies despite there being despite there being 16 times more interior edges than exterior edges.

When using best buddy accuracy as an estimation metric, this thesis differentiates between interior and exterior best buddy errors by giving interior best buddy errors higher weight.

4 Quantifying Solver Accuracy

Cho *et. al.* [13] defined the two metrics for quantifying the accuracy of a solver result namely: Direct Accuracy and Neighbor Accuracy. These metrics have been used in subsequent work [14, 15, 16, 17, 18]. This section describes the existing metrics, the weaknesses of these metrics, and proposes enhancements to these metrics to make them more meaningful for type 2 puzzles as well as when solving multiple puzzles simultaneously.

4.1 Direct Accuracy

Direct accuracy is the most naive accuracy measure. It is defined as the ratio of the number of pieces placed in the same location as the ground-truth (i.e. source). The formal definition of direct accuracy (DA) is shown in Equation (9) where n is the number of pieces in the source puzzle and c is the number of pieces placed in their correct location.

$$DA = \frac{c}{n} \quad (9)$$

Direct accuracy is vulnerable to shifts in the solved image where even a few misplaced pieces can cause a significant decrease in the accuracy. This can be particularly true when the ground-truth image’s puzzle dimensions are not known/fixed as described in Section 4.1.2.

This thesis proposes two modifications to the direct accuracy metric known as “Enhanced Direct Accuracy Score” and “Shiftable Enhanced Direct Accuracy Score”. They each described in the following two subsections; afterwards, the thesis describes why both metrics serve complementary roles.

4.1.1 Enhanced Direct Accuracy Score

The standard direct accuracy metric does not account for the possibility that in the solver output there may be pieces from multiple puzzles. For a puzzle P_i in the set of input puzzles P (where $P_i \in P$) and a set of solved puzzles S where S_j is a solved puzzle in S , then Equation 10 defines the Enhanced Direct Accuracy Score (EDAS).

$$EDAS_{P_i} = \arg \max_{s_j \in S} \frac{c_{i,j}}{n_i + \sum_{k \neq i} (m_{k,j})} \quad (10)$$

c_{ij} is the number of pieces from input puzzle P_i correctly placed (with no rotation in type 2 puzzles) in solved puzzle S_j while n_i is the number of pieces in puzzle P_i . $m_{k,j}$ is the number of pieces from an input puzzle P_k (where $k \neq i$) that are also in solved S_j .

When solving only a single puzzle, EDAS and the standard Direct Accuracy proposed by Cho *et. al.* are equivalent. In the case of solving multiple puzzles simultaneously, EDAS considers all pieces from the input puzzle of interest (P_i) while penalizing for pieces that are present from other input puzzles (P_k).

It is important note that EDAS is a score and not a measure of accuracy. While its value is bounded between 0 and 1 (inclusive), it is not specifically defined as the number of correct placements divided by the total number of placements since the denominator may be greater than the number of pieces in the solved puzzle S_j .

4.1.2 Shiftable Enhanced Direct Accuracy Score

As mentioned previously, direct accuracy is vulnerable to shifts in the original image. At times, the degree to which direct accuracy punishes shifts can be overly punitive.

Figure 3 shows an original image and an actual solver output. Note that only a single piece is misplaced, causing all pieces to be shifted over one location to the right. This single misplaced piece causes the direct accuracy score to drop to 0%. Had this same piece been misplaced at either the right or bottom side of the image, the direct accuracy would have been largely unaffected. The fact that direct accuracy can give such vastly differing results for essentially the same error indicates that it has room for improvement.



(a) Original Image



(b) Solver Output

Figure 3: Solver Output where a Single Misplaced Piece Catastrophically Affects the Direct Accuracy

Bibliography

- [1] A. D. Williams, *Jigsaw puzzles: an illustrated history and price guide*. Wallace-Homestead Book Co., 1990.
- [2] A. D. Williams, *The jigsaw puzzle: piecing together a history*. Berkley Books, 2004.
- [3] H. Freeman and L. Gardner, “Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition,” *IEEE Transactions on Electronic Computers*, vol. 13, pp. 118–127, 1964.
- [4] T. Altman, “Solving the jigsaw puzzle problem in linear time,” *Appl. Artif. Intell.*, vol. 3, pp. 453–462, Jan. 1990.
- [5] E. D. Demaine and M. L. Demaine, “Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity,” *Graphs and Combinatorics*, vol. 23 (Supplement), pp. 195–208, June 2007.
- [6] B. J. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, C. Dumas, S. Rusinkiewicz, and T. Weyrich, “A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 27, Aug. 2008.
- [7] M. L. David Koller, “Computer-aided reconstruction and new matches in the forma urbis romae,” *Bullettino Della Commissione Archeologica Comunale di Roma*, vol. 2, pp. 103–125, 2006.
- [8] S. L. Garfinkel, “Digital forensics research: The next 10 years,” *Digit. Investig.*, vol. 7, Aug. 2010.
- [9] T. S. Cho, M. Butman, S. Avidan, and W. T. Freeman, “The patch transform and its applications to image editing,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [10] L. Zhu, Z. Zhou, and D. Hu, “Globally consistent reconstruction of ripped-up documents,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 1–13, 2008.

- [11] W. Marande and G. Burger, “Mitochondrial dna as a genomic jigsaw puzzle,” *Science*, vol. 318, no. 5849, pp. 415–415, 2007.
- [12] Y.-X. Zhao, M.-C. Su, Z.-L. Chou, and J. Lee, “A puzzle solver and its application in speech descrambling,” in *Proceedings of the 2007 Annual Conference on International Conference on Computer Engineering and Applications*, pp. 171–176, World Scientific and Engineering Academy and Society (WSEAS), 2007.
- [13] T. S. Cho, S. Avidan, and W. T. Freeman, “A probabilistic image jigsaw puzzle solver,” in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR ’10, pp. 183–190, IEEE Computer Society, 2010.
- [14] D. Pomeranz, M. Shemesh, and O. 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, pp. 9–16, IEEE Computer Society, 2011.
- [15] G. Paikin and A. 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.
- [16] A. 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, pp. 382–389, IEEE Computer Society, 2012.
- [17] D. Sholomon, O. David, and N. S. Netanyahu, “A genetic algorithm-based solver for very large jigsaw puzzles,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pp. 1767–1774, 2013.
- [18] K. Son, J. Hays, and D. B. Cooper, “Solving square jigsaw puzzles with loop constraints,” in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, pp. 32–46, 2014.