This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

# Solving Small-piece Jigsaw Puzzles by Growing Consensus

Kilho Son[1,*] Daniel Moreno[1] James Hays[2] David B. Cooper[1]
Brown University[1] Georgia Institute of Technology[2]
kilho.son@gmail.com*

## Abstract

*In this paper, we present a novel computational puzzle solver for square-piece image jigsaw puzzles with no prior information such as piece orientation, anchor pieces or resulting dimension of the puzzle. By "piece" we mean a square dxd block of pixels, where we investigate pieces as small as 7x7 pixels. To reconstruct such challenging puzzles, we aim to search for piece configurations which maximize the size of consensus (*i.e. *grid or loop) configurations which represent a geometric consensus or agreement among pieces. Pieces are considered for addition to the existing assemblies if these pieces increase the size of the consensus configurations. In contrast to previous puzzle solvers which goal for assemblies maximizing compatibility measures between all pairs of pieces and thus depend heavily on the pairwise compatibility measure used, our new approach reduces the dependency on the pairwise compatibility measures which become increasingly uninformative at small scales and instead exploits geometric agreement among pieces. Our contribution also includes an improved pairwise compatibility measure which exploits directional derivative information along adjoining boundaries of the pieces. For the challenging unknown orientation piece puzzles* where the size of pieces is small*, we reduce assembly error by up to* 75% *compared with previous algorithms for standard datasets.*

## 1. Introduction

Recent advances in computer science and engineering led to computational algorithms to solve digitized jigsaw puzzles including images [4], broken ancient articles [22, 16], documents [26], photographs [13] and separated, fractured bones [14]. The computational puzzle solver has been an important tool to assist the time-consuming assembly of otherwise infeasible jigsaw puzzles. Many state of the art algorithms perform beyond human ability and thus play essential roles in discovering unrevealed objects from many fragments.

Introduced by Freeman *et al*. [8], the task of solving *square* jigsaw puzzles has been a challenge to many researchers. Many previous works [1, 4, 25, 18, 9, 7, 20, 23, 17] tackle non-overlapping square piece image jigsaw puzzles and notable progress has been made. In addition, researchers have demonstrated the utility of these methods in applications such as seamless image editing via a bag of patches [3], reconstructing a new image from a bag of patches for image retrieval and recognition [12], and matching templates [5].

In this paper, we propose a solver for challenging square piece image jigsaw puzzles where no prior information – piece orientation, anchor pieces (ground truth configurations), and dimension of resulting puzzles (i.e number of squares in the horizontal and vertical directions) are all unknown. Furthermore, we also attempt to assemble puzzles with small pieces, 14 by 14, 10 by 10 and 7 by 7 pixels, in contrast to the more common 28 by 28 pixel benchmark. When the size of a piece becomes smaller, the information in the pieces is reduced so that the predefined compatibility measure is no longer reliable enough to configure the puzzles correctly. Thus, we must develop smarter and more robust strategies to assemble small size pieces successfully.

For assembling these demanding puzzles, we choose a framework that estimates pairwise matching and subsequently aligns these matched pieces by making use of all the pairwise costs involved in the aligned sets. Since we are using this framework, our puzzle solver is capable of simultaneously reconstructing multiple puzzles, whose pieces are mixed together, with no prior information. Our solver also handles missing fragments, which is a common situation in archaeological applications [24]. The most challenging aspect of this puzzle reconstruction strategy is that the consecutive pairwise matching should not include false pairwise matches. Even a single initial false pairwise match may lead to a entire incorrect configuration.

To overcome these issues, we contribute two crucial steps in the matching-based assembly algorithm: an assembly strategy and a pairwise compatibility measure. For assembling pieces, we consider that the pairwise compatibility measure could be noisy when the pieces are small, the pieces are from textureless regions, or the boundaries of pieces are aligned with boundaries of man-made structures.

Thus, we exploit an idea that an assembly by multiple modest bonds is more reliable than an assembly by a strong single bond which is found by the predefined pairwise compatibility measure. From this idea, we propose a new objective for assembling puzzles, specifically maximizing consensus (grid or loop) configurations. In order to search for configurations which maximize consensus, we add a new piece to the existing grid configurations if the addition increases the size of grid configurations regardless of pairwise compatibility measure. In addition, we also improve the pairwise compatibility measure by adding derivative information along the boundary in the piece to the pairwise compatibility measure. Our proposed algorithm reduces assembly error by up to $75\%$ compared with previous algorithms for the challenging unknown orientation puzzles from standard datasets where the size of pieces is small.

## 1.1. Related Works

Although Demaine *et al*. [6] discovered that puzzle assembly is an NP-hard problem, many published algorithms incrementally improved the performance with novel ideas. For non-overlapping square piece jigsaw puzzles, many researchers have studied solvers to enhance reconstruction accuracy and reduce prior information needed. Earlier works [1, 4, 25, 18, 7, 20] solved image puzzles where the orientation of the puzzle pieces is known, and additional information is given such as the dimensions of resulting puzzles. The positions of the puzzle pieces are the only unknowns. If the dimensions of the resulting puzzle are known, the puzzle problem can be formulated as recovering the optimal 2D permutation of labels where the labels are the IDs of the pieces. Implicitly or explicitly, their objectives are to maximize a predefined pairwise compatibility measure between neighbor pieces and various optimization methods were applied such as a belief propagation [4], particle filtering [25], greedy algorithms [18], constrained quadratic function minimization [9] and genetic algorithms [20]. Over the years these methods substantially enhanced reconstruction accuracy, increased the number of puzzle pieces that the solver can perfectly assemble, and reduced computational time. However, when their compatibility measures are not reliable enough, their objective functions lead to false configurations.

Gallagher [9] introduced a new type of image puzzle where the orientation of the piece is also unknown with no prior information such as a resulting puzzle dimension. Since the dimension of the puzzle is unknown, Gallagher formulated the puzzle problem as optimally linking the pieces without geometric conflicts such as piece overlaps and solved it with Kruskal's algorithm [11]. Gallagher also proposed a powerful pairwise compatibility measure, Mahalanobis Gradient Compatibility (MGC) which penalizes changes across the pieces when they are more than expected and normalizes them with sample covariance estimated by pixels around the edge of the piece. Paikin *et al*. [17] also proposed a similar greedy assembly strategy with a more careful initial configuration. These assembly strategies [9, 17] are also based on a strong belief on the possibly incorrect pairwise compatibility measure. They accept a puzzle configuration as long as there is a single strong bond even though the other sides of these pieces are incompatible with their neighbors. Thus, when false pair matches return high compatibility values or true pair matching returns low compatibility value, their assembly strategies fail to build true assemblies.

Son *et al*. [23] also proposed a solver for unknown orientation piece puzzles with no prior knowledge. They discovered that if pairwise compatible pieces form a loop, the pairwise matches are more reliable than for pieces that do not form a loop. Based on this idea, they assembled puzzles by recursively building loops of loops (e.g. 4 overlapping 2x2 loops into a 3x3 wide loop). Son *et al*. [23] also relies on a predefined compatibility measure which may be possibly incorrect. They search for configurations where a piece is compatible with "all" of its neighbors. If the pairwise compatibility measure misses even a single true neighbor, the algorithm loses many true configurations and fails to find true assemblies.

Different from the previous algorithms, our assembly strategy depends less on the predefined pairwise compatibility measure used. When a pairwise compatibility measure returns high values on false pairs, we do not accept these pairs unless these pairs agree with the other neighbor pieces. Conversely, when true pairs output low compatibility measure values, we accept these configurations if these pairs receive agreements from the other neighbors pieces. By exploiting consensus information from the neighbors, our proposed algorithm is robust to noisy compatibility measure values which occur in many challenging puzzles with small size pieces and/or non-informative pieces.

"Consensus" information has been utilized in many fields. Robust estimation, exploiting consensus of the measurements such as RANSAC has presented impressive improvement on many computer vision applications [10, 19]. The robust estimation has proven its robustness for camera geometry estimation [21]. The consensus information was used in low-level vision [2]. Chakrabarti *et al*. introduced a multi-scale framework to accumulate information from all the scales and found a consensus to reject the outlier information. The consensus concept, "Best Buddy" was introduced for solving the jigsaw puzzles by Pomeranz *et al*. [18]. They discovered that if both pieces agree that they are the best neighbors, the matching is more reliable.

[2] The assemblies in the bag of consensus are always square, and N by N consensus configuration means that the assembly is square with side dimension N. We also refer to the number N as the size of the configuration.

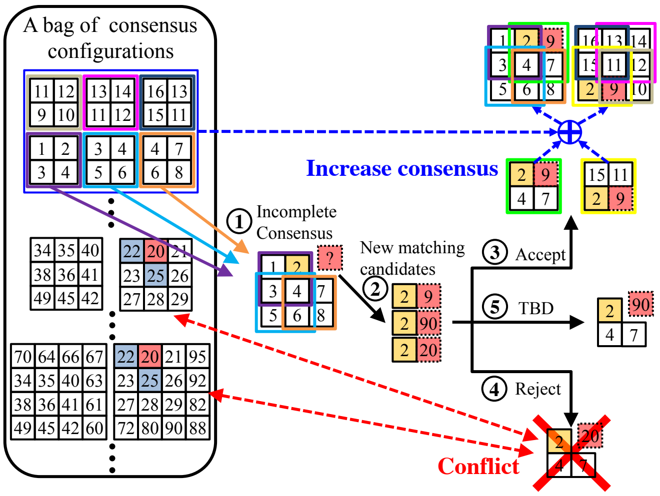



Figure 1. **An exemplary single iteration of increasing consensus configurations.** The squares represent pieces and the numbers their unique IDs. From the bag of consensus configurations, ① we search for incomplete 3 by 3 [2] consensus from three of 2 by 2 consensuses and ② find pairwise matching candidates to complete this 3 by 3 consensus. Since top of piece 7 matches with bottom of pieces 9, 90 and 20, the right side of piece 2 to the left side of those pieces are considered as new pairwise matching candidates. Likewise, pairwise matches with right side of piece ID 2 could propose more pairwise matching candidates with piece ID 7. ③ If the pairwise matching completes at least two of 2 by 2 consensus configurations, we accept the pairwise matching. If the pairwise matching is true, this addition sometimes generates 2 or more additional bigger size of consensus configurations. In this example, two of 3 by 3 consensus configurations are generated additionally. ④ If the candidate match conflicts with existing larger consensus configurations (in this example, 3 by 3 and 4 by 4 configurations), we reject the pairwise match. Otherwise, ⑤ we hold off the decision to the next iteration. We note that the decision is made only by existing consensus configurations in the bag, independent of the pairwise compatibility strength.

## 2. Puzzle Assembly by Growing Consensus

Prior works [4, 18, 25, 7, 9, 20, 23] present methods to search for assemblies based on predefined pairwise compatibility measures. These algorithms perform accurately on puzzles where the pairwise compatibility measure exactly identifies which pieces are true neighbors. If the pairwise compatibility measure is inaccurate the algorithms often fail. These cases happen frequently when the size of the pieces is small, the textures on the pieces have no distinguishing characteristics, or the boundaries of the pieces are aligned with edges of man-made structures. Here, we propose a new objective for puzzle assembly – maximizing consensus configurations. In contrast to previous algorithms, we attempt to find configurations which maximize a geometric consensus between pieces. *i.e.* We believe harmonious (non-conflicting and supporting each other) piece configurations are more important than a single bond between two pieces defined by the pairwise compatibility measure. With the proposed objective, we reduce the dependency on the possibly erroneous pairwise compatibility measure and instead exploit the geometric information of existing piece configurations. In the case where the compatibility measure fails to distinguish the true neighbor pieces, we successfully assemble the puzzle with help of geometric consensus from neighbor pieces.

**Outline.** From a predefined pairwise compatibility measure, we initially generate a bag of consensus configurations (Section 2.1). Iteratively, we propose new pairwise matching candidates which possibly increase the size of the consensus bag (Section 2.2), and determine if we accept the candidate or not (Section 2.3). See Figure 1 and Algorithm 1 for detailed illustration.

### 2.1. Initialization

We first search for initial pairwise matching candidates. From the pairwise compatibility measure, dissimilarity values are calculated for all combinations of two pieces. For puzzles with unknown piece orientation, there exist 16 different relative poses between two pieces. When orientation is known, there exist 4 different configurations between two pieces. For each side of the pieces, we divide the dissimilarity values with other pieces by the second smallest dissimilarity value to output relative dissimilarity as proposed by Gallagher [9]. If a pair presents the relative dissimilarity value smaller than a threshold (1.07), we consider it as a pairwise matching. Each side of a piece is allowed to have a maximum of 10 candidate matches.

From the pairwise matching candidates, we generate a bag of consensus assemblies. Following Son *et al.* [23], we discover cycles (loops) of four pairwise matches resulting in assemblies of 2 by 2 pieces. Once we find pairwise matching of 2 by 2 super pieces, we perform the same process as before to generate 3 by 3 super pieces from four 2 by 2 super pieces. More generally, with this process we generate N+1 by N+1 super pieces from four N by N super pieces. We iteratively increase the size of super pieces, one row and one column at a time, until we find the maximum size super pieces for the puzzle. We refer the entire set of generated super piece from size 2 to maximum as "a bag of consensus configurations". As noted by Son *et al.* [23], as the consensus configurations become larger, the correctness of the assemblies grows.

### 2.2. Proposing New Pair Matching Candidates

In this step, we discover new pairwise matching candidates which are not found by the pairwise compatibility measure in the initial step. The new pair matching candidate should possibly increase size of consensus configurations in the bag. For this purpose, we examine from the largest to the smallest consensus configurations in the bag, and search

**Algorithm 1:** Growing consensus

- Initialize a Bag of Consensus Configurations (BCC)
**while** *Incomplete consensus exists in BCC* **do**
    - Find the largest and unvisited incomplete consensuses from 3 consensus config. in BCC
    - Find new candidate pairwise matches to complete the incomplete consensuses
    **for** *all the new candidate pairwise matches* **do**
        - Determine Accept, Reject or TBD
        - Accept: Construct new consensus configurations and add them to BCC
        - Reject : Discard the pairwise matches
        - TBD : Defer the decision to next iteration
    **end**
**end**
**return** *BCC*

for incomplete consensus from three of same size (order) of consensus configurations in the bag. When three of N by N super pieces are matched well but only one N by N super piece is missing so that they cannot form a N+1 by N+1 consensus, we call them an incomplete consensus. If a single pairwise matching completes the incomplete consensus as exemplified in Figure 1, we count it as a new pairwise matching candidate. We do not accept the new pairwise matching candidate now since the matching received insufficient agreement from its neighbor pieces.

### 2.3. Rejecting or Accepting

For the each new pairwise matching candidate, we take action of rejection, acceptance, or To Be Determined (TBD). A pairwise matching candidate is rejected if the pairwise matching is geometrically inconsistent with existing consensus configurations whose size is greater than size of consensus configurations that are used for generating the new pairwise matching candidates. We accept a pairwise matching candidate as a true matching if the pairwise matching candidate generates 2 or more new consensus configurations. Otherwise, we consider the pairwise matching candidates as TBD which will be determined in the next iteration. That is, if the new pairwise matching candidates neither conflict with current configurations nor contribute increase size of consensus, we leave the decision to the next iteration. We note that the decision is made only by the current consensus configurations in the bag and is independent of pairwise compatibility measure.

### 2.4. Merging, Trimming and Filling

We merge consensus configurations in the bag based on the overlapped pieces. If two configurations conflict we remove the smaller one. If they are the same size average pairwise compatibility measures in the consensuses are used as a tiebreaker. The resulting assemblies from these steps are not guaranteed to be a rectangular shape, since the proposed algorithm is a matching-based algorithm whose results are free-formed. We adopt the trimming and filling steps to make the result rectangular from Gallagher [9].

## 3. Validations of Maximizing Consensus

To validate our new proposed assembly algorithm, we show that a number of positive neighbor matches (consensus from neighbor pieces) is more important than a single strong positive match for a correct piece configuration. When a compatibility measure between a piece of interest and a adjacent piece is smaller than a threshold, we refer that the neighbor as positive and otherwise as negative.

We first calculate a probability that a piece configuration is correct where $\epsilon$ number of the neighbors are negatives and the others are positives. Let us define events,

$$\begin{aligned} &M_p\text{: A pairwise piece configuration is positive.}\\ &M_n\text{: A pairwise piece configuration is negative.}\\ &M_t\text{: A pairwise piece configuration is true.}\end{aligned} \tag{1}$$

A true pairwise piece configuration means a ground truth configuration that we goal for. Let us assume that 1) all the pairwise piece configurations are independent and 2) share the same conditional probabilities, $P(M_t|M_n)$, $P(M_t|M_p)$. When we find a piece configuration which has $\epsilon$ number of negative neighbors and $4-\epsilon$ number of positive neighbors, the probability that this configuration is true is,

$$P(M_t|M_n)^{\epsilon}P(M_t|M_p)^{(4-\epsilon)}. \tag{2}$$

Now, we summarize the error of pairwise matching candidates from the initial step of our algorithm by two metrics, precision $\beta$ (ratio of true positives to positives) and recall $\alpha$ (ratio of true positives to trues). Let's set $\phi$ as a number of positive pairs and $\nu$ as a number of negative pairs. Then, conditional probabilities are estimated by counting numbers of false negative matches and true positive matches,

$$P(M_t|M_n) = \beta\frac{\phi}{\nu}\frac{1-\alpha}{\alpha}, \qquad P(M_t|M_p) = \beta. \tag{3}$$

Thus, the probability in the Equation 2 becomes

$$P(M_t|M_n)^{\epsilon}P(M_t|M_p)^{(4-\epsilon)} = \beta^4(\frac{\phi}{\nu})^{\epsilon}(\frac{1-\alpha}{\alpha})^{\epsilon}. \tag{4}$$

We find 10 positive pairwise matches maximally for each side of the pieces so that the number of positive pairs are

$$\phi = O(\#pieces), \tag{5}$$

where $\#pieces$ represents a number of puzzle pieces. And, the other possible pairwise piece configurations are all negative so that the number of negative pairs are

$$\nu = O(\#pieces^2). \tag{6}$$

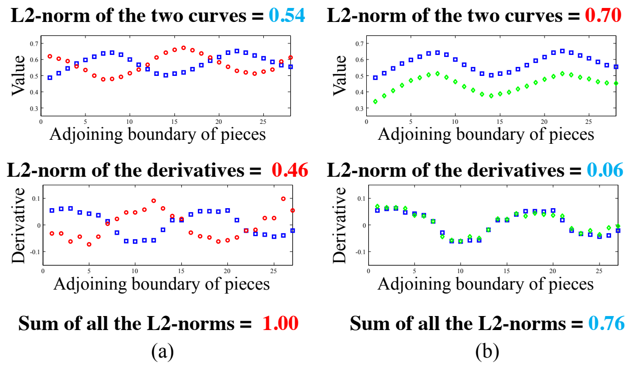



Figure 2. **Importance of derivative information for defining dissimilarity between pieces.** Defining dissimilarity between two pieces can be thought as defining dissimilarity between two discretized curves where the two curves represent the values along the boundaries of the two pieces. Prior works define $L2$ norm as dissimilarity of two curves. In (a), two curves are more dissimilar than two curves in (b). But, the $L2$ norm of the two curves in (b) is greater than that in (a) which is not desired. The $L2$ norm of the two curves does not exactly represent the dissimilarity between two curves. By considering the derivatives in a direction along the boundary, we can compensate the erroneous measure. The value of the $L2$ norm of the derivative curve error in (a) is greater than that in (b) which is correct. For this reason, we incorporate derivative information along the adjoining boundaries for defining pairwise compatibility measure.

With the fixed and moderate $\alpha$ and $\beta$ ranging from $0.1 \sim 0.9$, as a piece configuration includes more negative neighbors, the piece configuration drastically reduces its probability to be a correct configuration due to the term $(\frac{\phi}{\nu})^{\epsilon}$ in the Equation 4.

If we think that one of the 4 neighbors is a strong positive *i.e.* $P(M_t|M_p) \approx 1$ ,and the others are negatives, the probability in the Equation 2 becomes

$$P(M_t|M_n)^3 P(M_t|M_p) = \beta^3 (\frac{\phi}{\nu})^3 (\frac{1-\alpha}{\alpha})^3 \beta', \quad (7)$$

where $\beta'$ is almost 1. Because this probability is already very small due to the term $(\frac{\phi}{\nu})^3$, the $\beta'$ does not strongly affect the probability. This represents a single neighbor bond does not heavily increase the probability to be a correct configuration.

## 4. Pairwise Compatibility Measure

For measuring compatibility between two pieces, we calculate the dissimilarity between adjoining boundary pixel values of the two pieces. Our pairwise compatibility measure is constructed by two ideas. We first consider pixel intensity changes crossing the border between two pieces. If the changes are greater or smaller than expected, we penalize it. This is similar to Mahalanobis Gradient Compatibility (MGC) by Gallagher [9] but we estimate the expected changes more accurately than is done in MGC. Secondly, we also incorporate directional derivative information along the adjoining boundaries of the pieces for defining the pairwise compatibility measure. If the directional derivative changes across the pieces are greater or smaller than expected, we also penalize it. The proposed pairwise compatibility measure is robust to the illumination changes on the boundary pixels that result in different offset levels because the derivative information is invariant to offset of pixel values as explained in Figure 2.

Our proposed pairwise compatibility measure of a piece $x_i$ on the left side of $x_j$ consists of four terms,

$$\begin{aligned} C_{LR}(x_i, x_j) =& D_{LR}(x_i, x_j) + D_{RL}(x_i, x_j) \\ &+ D'_{LR}(x_i, x_j) + D'_{RL}(x_i, x_j). \end{aligned} \quad (8)$$

The first two terms penalize larger or smaller changes of the pixel value across the two pieces than expected. When the size of piece is $P$,

$$D_{LR}(x_i, x_j) = \sum_{p=1}^{P} (\Lambda^{ij}_{LR}(p) - E^{ij}_{LR}(p)) V^{-1}_{iL} (\Lambda^{ij}_{LR}(p) - E^{ij}_{LR}(p))^T, \quad (9)$$

where,

$$\Lambda^{ij}_{LR}(p) = x_j(p, 1) - x_i(p, P), \quad (10)$$

$$E^{ij}_{LR}(p) = \frac{1}{2}(x_i(p, P) - x_i(p, P-1) + x_j(p, 2) - x_j(p, 1)). \quad (11)$$

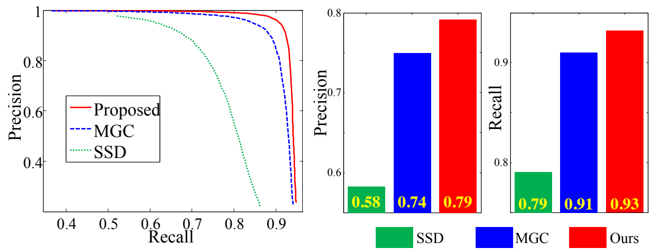
$x_i(k_1, k_2)$ is a 3 dimensional vector representing red, green and blue pixel intensities at the position $(k_1, k_2)$ in the puzzle piece $i$. $\Lambda^{ij}_{LR}(p)$ is a pixel intensity change across the boundary of the two pieces and $E^{ij}_{LR}(p)$ is a expected change across the boundary of the two pieces. We normalize the changes with the sample covariance $V_{iL}$ calculated from samples, $\{x_i(p, P) - x_i(p, P-1))|p = 1, 2, \ldots P\}$. $D_{RL}(x_i, x_j)$ is calculated by the same way.

$D'_{LR}(x_i, x_j)$ is calculated with the similar manner. The only difference is that we substitute the pixel values $x_i(k_1, k_2)$ in the Equations 9, 10 and 11 with directional derivative of pixel values along the boundary of the piece,

$$\delta_i(k1, k2) = x_i(k1, k2) - x_i(k1-1, k2). \quad (12)$$

And, the sample variance $V_{iL}$ for $D'_{LR}(x_i, x_j)$ is calculated from samples, $\{\delta_i(p, P) - \delta_i(p, P-1))|p = 2, \ldots P\}$.

MGC assumes that the expected change across the piece boundaries is an average difference between the final two column in $x_i$ whereas we estimate the expected changes as an average of local pixel changes in both pieces. Our expected changes are more accurate than those used in MGC especially when the size of the piece is large.



(a) (b)

Figure 3. **Pairwise compatibility measure performance.**

## 5. Experiments

We performed experiments with many standard image datasets from Cho *et al.* [4] (MIT dataset), Olmos *et al.* [15] (Mcgill dataset) and Pomeranz *et al.* [18] (Pomeranz 805, 2360 and 3300 dataset). The MIT, Mcgill and Pomeranz 805 dataset each include 20 images and the Pomeranz 2360 and 3300 dataset each contain 3 large images.

### 5.1. Pairwise Compatibility Measure Performance

For unknown orientation puzzles, a strict accuracy metric is inappropriate to compare the performance of pairwise compatibility measures because the number of true negatives is far larger than the number of positives and false negatives. Chance is vanishingly small and thus accuracy tends to be small in absolute terms. Thus, the accuracy metric does not distinctly show the performance difference between the algorithms. One approach to evaluate the performance of pairwise compatibility measure is precision and recall [1].

We used the MIT dataset and generate challenging unknown orientation puzzles where the piece size is P=28 pixels and the number of pieces is K=432 to evaluate performance of compatibility measures. In Figure 3 (a), precision and recall curves for MGC [9], Sum of Squared Distance(SSD) [4] and the proposed pairwise compatibility measure are presented. In Figure 3 (b), We also calculate precision and recall values when each side of a piece matches with a single piece with lowest dissimilarity value. Our method reduces precision and recall error by 20% compared to MGC. Furthermore, we performed component analysis of our proposed pairwise compatibility measure with the same setup. Figure 4 shows that our method performs better than the MGC [9] without derivative information. And, with the derivative information, the performance is even more enhanced.

### 5.2. Unknown Orientation Piece Puzzles

We compared our algorithm with previous works [9, 23, 17] for solving the most challenging unknown orientation piece puzzles. For evaluation, we utilized metrics from Cho *et al.* [4] and Gallagher [9]. "Direct Comparison" measures a percentage of pieces whose absolute position

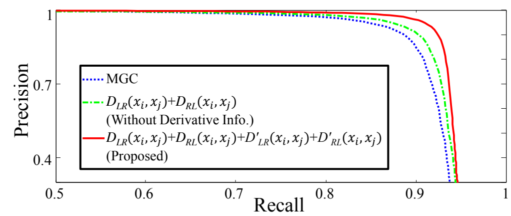


Figure 4. **Component analysis of compatibility measure.**

| | Direct | Neighbor | Largest | Perfect |
|---|---|---|---|---|
| Gallagher [9] | 82.2% | 90.4% | 88.9% | 9 |
| Son *et al.* [23] | 94.7% | 94.9% | 94.6% | **12** |
| Paikin *et al.* [17] | 95.4% | 95.4% | - | - |
| Proposed | **95.9%** | **95.6%** | **95.9%** | **12** |

Table 1. **Reconstruction performance on unknown orientation piece puzzles from the MIT dataset.**

| | Mcgill dataset(K=540, P=28) | | | |
|---|---|---|---|---|
| | Direct | Neighbor | Largest | Perfect |
| Gallagher [9] | 72.0% | 73.3% | 72.8% | 7 |
| Son *et al.* [23] | 89.1% | 92.5% | 89.0% | 10 |
| Proposed | **91.4%** | **94.5%** | **91.4%** | **11** |
| | Pomeranz dataset(K=805, P=28) | | | |
| | Direct | Neighbor | Largest | Perfect |
| Gallagher [9] | 83.2% | 85.5% | 83.5% | 5 |
| Son *et al.* [23] | 86.4% | 88.8% | 86.3% | **10** |
| Proposed | **88.7%** | **93.3%** | **88.7%** | 9 |
| | Pomeranz dataset(K=2360, P=28) | | | |
| | Direct | Neighbor | Largest | Perfect |
| Gallagher [9] | 60.5% | 62.5% | 60.8% | 0 |
| Son *et al.* [23] | 94.0% | **95.3%** | **94.0%** | 0 |
| Proposed | **94.1%** | 95.2% | **94.0%** | **1** |
| | Pomeranz dataset(K=3300, P=28) | | | |
| | Direct | Neighbor | Largest | Perfect |
| Gallagher [9] | 80.3% | 81.9% | 80.2% | **1** |
| Son *et al.* [23] | **89.9%** | 93.4% | **89.9%** | **1** |
| Proposed | 89.5% | **95.3%** | 89.5% | **1** |

Table 2. **Reconstruction performance on unknown orientation piece puzzles from the Mcgill and the Pomeranz dataset.**

and orientation are both correct. And if "Direct Comparison" is 100%, "Perfect Reconstruction" is 1 and otherwise 0. "Neighbor Comparison" is a percentage of true positive pairwise matches. "Largest Component" is the area of the largest group of correctly assembly parts.

**MIT dataset:** We first verified that the proposed assembly algorithm increases the size of consensus configurations. Figure 6 shows largest size (*i.e.* order or dimension) of consensus assemblies for each image puzzle (unknown orientation piece puzzle, P=28, K=432) from MIT dataset before and after our proposed assembly algorithm. We observed that our assembly algorithm increases the size of largest consensus assemblies and reaches the maximum possible size of consensus assembly for most of image puzzles.

Figure 5 qualitatively compares assembly results on unknown orientation piece puzzles from previous works and

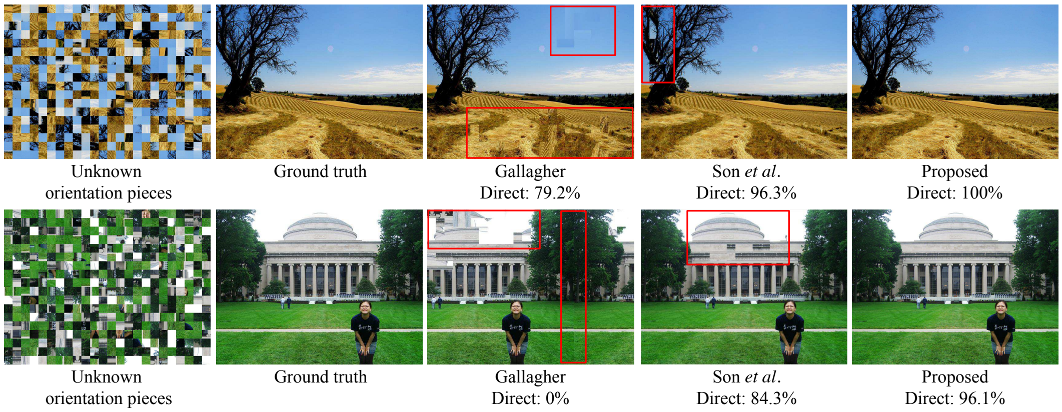



Figure 5. **Qualitative reconstruction performance on unknown orientation piece puzzles.** The prior algorithms return false configurations in the red rectangle regions whereas our proposed algorithm correctly assembles those challenging pieces. Refer to the supplemental material for more qualitative results. https://sites.google.com/site/kilhoson/puzzle2d_mc

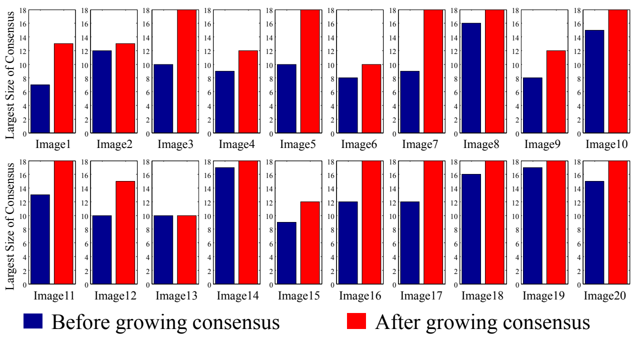



Figure 6. **Analysis of proposed assembly algorithm.**

our algorithm. The size of each piece is P=28 pixels and the number of pieces in a puzzle is K=432. These images in Figure 5 are very challenging images from the MIT dataset due to lack of texture or man-made structures aligned with boundary of puzzle pieces. The other algorithms [9, 23] struggled to correctly assemble the puzzles (See the red rectangles in Figure 5) whereas our algorithm successfully assembled the demanding image puzzles. We note that our result of the second image puzzle in Figure 5 is visually perfect but the direct comparison is not 100% due to the saturated pieces where all the pixel values are maximum so that no information is included in the pieces. We performed quantitative analysis with the same setup (Table 1). Our method reduces reconstruction error from the prior algorithms. We note that the MIT dataset includes 5 images where parts of the images are saturated such as the second image in the Figure 5. Due to these pieces, it is not possible to reach to 100% reconstruction accuracy.

**Mcgill and Pomeranz dataset:** We also tested our algorithm with Mcgill dataset [15] and Pomeranz dataset [18]. For images from the Mcgill, Pomeranz805, Pomeranz2360 and Pomeranz3300, we generated K = 540, 805, 2360, 3300 numbers of puzzle pieces, respectively, where the orientation of pieces is unknown and the size of the pieces is P=28. Our algorithm outperforms the previous works (Table 2).

**Evaluation on various size and number of pieces:** As proposed by Gallagher [9], we performed more experiments by varying the size of pieces P=7,10,14,28 pixels and the number of puzzle pieces K=221, 432, 1064 using the MIT dataset. When the size of a piece is small, each piece includes less information so that the compatibility measure is unreliable. Thus, we can evaluate the robustness of the assembly algorithms with weaker guidance from the pairwise compatibility measure. The Figure 7 shows the reconstruction performance comparisons for each case. It is notable that our proposed method significantly reduced the reconstruction error by up to 75% from Son *et al.* [23] when the size of a piece is P=14. In fact, our proposed algorithm almost maintained the reconstruction performance although the size of a piece was reduced to P=14 whereas the previous works [9, 23] drastically reduced their reconstruction performance.

**Algorithm component analysis:** We analyzed the contribution of each step in our algorithm. We used two pairwise compatibility measures, MGC [9] and our pairwise compatibility measure (refer to DC), and two assembly strategies, tree-based [9] and our assembly algorithm (refer to consensus-based). From the methods, we formed 4 different assembly algorithms from all combinations of the algorithms. Figure 8 shows the reconstruction performance comparison on an unknown orientation piece puzzle from images from the MIT dataset (P=28, K=432). Each step of our method contributes the performance improvement.

Our algorithm solves puzzles where the pieces are from multiple images and some pieces are missing (Figure 9).

### 5.3. Known Orientation Piece Puzzles

We compared our algorithm with the previous works [18, 7, 20, 23, 17] when orientation of each piece is known and only position of piece is unknown in Table 3. Our perfor-

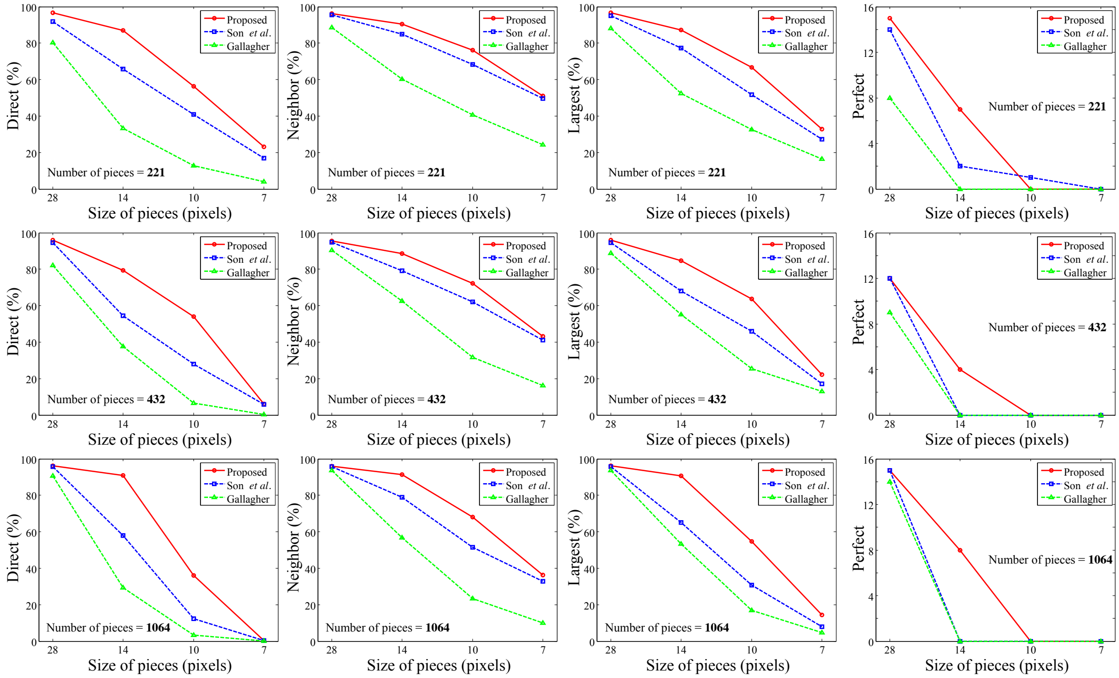



Figure 7. **Reconstruction performance of unknown orientation piece puzzle with various piece size and number of pieces.**

| | MIT(432 Pieces) | | Mcgill(540 Pieces) | | Pomeranz(805 Pieces) | | Pomeranz(2360Pieces) | | Pomeranz(3300Pieces) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Direct | Neighbor | Direct | Neighbor | Direct | Neighbor | Direct | Neighbor | Direct | Neighbor |
| Pomeranz *et al.* [18] | 94.0% | 95.0% | 83.5% | 90.9% | 80.3% | 89.7% | 33.4% | 84.7% | 80.7% | 85.0% |
| Andalo *et al.* [7] | 91.7% | 94.3% | 90.6% | 95.3% | 82.5% | 93.4% | - | - | - | - |
| Sholomon *et al.* [20] | 86.2% | **96.2%** | 92.8% | 96.0% | **94.7%** | **96.3%** | 85.7% | 88.9% | 89.9% | 92.8% |
| Son *et al.* [23] | 95.6% | 95.5% | 92.2% | 95.2% | 93.1% | 94.9% | **94.4%** | 96.4% | 92.0% | **96.4%** |
| Paikin *et al.* [17] | **96.2%** | 95.8% | 93.2% | **96.1%** | 92.5% | 95.1% | 94.0% | 96.3% | 90.6% | 95.3% |
| Proposed | 95.4% | 95.5% | **93.6%** | **96.1%** | 92.8% | 95.0% | 94.1% | **96.7%** | **94.9%** | 95.1% |

Table 3. **Reconstruction performance on known orientation piece puzzles from the Mcgill and Pomeranz dataset.** P=28, K=432.

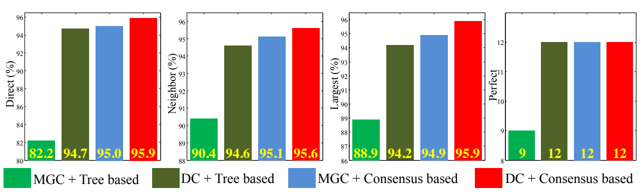



Figure 8. **Algorithm component analysis.** DC and consensus-based refer to our proposed pairwise compatibility measure and assembly strategy respectively.

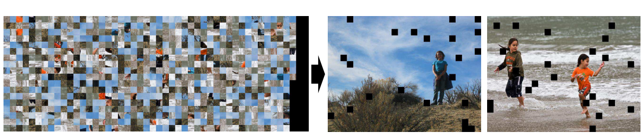


Figure 9. **Mixed, missed and unknown orientation piece puzzle.** The size of pieces is P=28. Puzzles are from two images and 5% of pieces are missing.

mance on the known orientation piece puzzle is comparable with those of previous works.

Our algorithm is implemented in Matlab and takes 120 seconds for assembling 432 pieces on a modern PC.

## 6. Conclusion

We present a novel computational puzzle solver for square piece image jigsaw puzzles with no prior information such as piece orientation, anchor pieces, or resulting dimension of the puzzle. When the size of pieces becomes smaller so that pairwise compatibility measures become unreliable, our algorithm achieves much higher reconstruction accuracy than state of the art methods. The strength of our algorithm derives from its reduced dependency on a pairwise compatibility measure, which is often erroneous, and instead exploits geometric information from reliable neighbor configurations for puzzle assembly. In addition, a more accurate pairwise compatibility measure which utilizes directional derivative information along adjoining boundaries of the pieces is proposed and used.

## References

[1] N. Alajlan. Solving square jigsaw puzzles using dynamic programming and the hungarian procedure. *American Journal of Applied Science*, 5(11):1941 – 1947, 2009. 1, 2, 6

[2] A. Chakrabarti, Y. Xiong, S. J. Gortler, and T. Zickler. Low-level vision by consensus in a spatial hierarchy of regions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[3] T. S. Cho, S. Avidan, and W. T. Freeman. The patch transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8), 2010. 1

[4] T. S. Cho, S. Avidan, and W. T. Freeman. A probabilistic image jigsaw puzzle solver. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 2, 3, 6

[5] T. Dekel, S. Oron, S. Avidan, M. Rubinstein, and W. Freeman. Best buddies similarity for robust template matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. 1

[6] E. D. Demaine and M. L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23 (Supplement), June 2007. 2

[7] S. G. Fernanda A. Andal, Gabriel Taubin. Solving image puzzles with a simple quadratic programming formulation. In *Conference on Graphics, Patterns and Images*, 2012. 1, 2, 3, 7, 8

[8] H. Freeman and L. Garder. Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. *Electronic Computers*, 13:118–127, 1964. 1

[9] A. C. Gallagher. Jigsaw puzzles with pieces of unknown orientation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 3, 4, 5, 6, 7

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 2

[11] J. Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *American Mathematical Society*, 1956. 2

[12] H. Kato and T. Harada. Image reconstruction from bag-of-visual-words. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[13] H. Liu, S. Cao, and S. Yan. Automated assembly of shredded pieces from multiple photos. In *ICME*, 2010. 1

[14] P. Liu. *A system for computational analysis and reconstruction of 3d comminuted bone fractures*. PhD thesis, The University of North Carolina, 2012. 1

[15] A. Olmos and F. A. A. Kingdom. A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception*, 2004. 6, 7

[16] G. Oxholm and K. Nishino. Reassembling thin artifacts of unknown geometry, 2011. International Symposium on Virtual Reality, Archaeology and Cultural Heritage. 1

[17] G. Paikin and A. Tal. Solving multiple square jigsaw puzzles with missing pieces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 6, 7, 8

[18] D. Pomeranz, M. Shemesh, and O. Ben-Shahar. A fully automated greedy square jigsaw puzzle solver. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1, 2, 3, 6, 7, 8

[19] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm. Usac: A universal framework for random sample consensus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):2022–2038, Aug 2013. 2

[20] D. Sholomon, O. David, and N. Netanyahu. A genetic algorithm-based solver for very large jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 3, 7, 8

[21] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, July 2006. 2

[22] K. Son, E. B. Almeida, and D. B. Cooper. Axially symmetric 3d pots configuration system using axis of symmetry and break curve. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1

[23] K. Son, J. Hays, and D. Cooper. Solving square jigsaw puzzles with loop constraints. In *Computer Vision ECCV 2014*, Lecture Notes in Computer Science. Springer International Publishing, 2014. 1, 2, 3, 6, 7, 8

[24] A. Willis and D. B. Cooper. Computational reconstruction of ancient artifacts. *IEEE Signal Processing Magazine*, pages 65–83, 2008. 1

[25] X. Yang, N. Adluru, and L. J. Latecki. Particle filter with state permutations for solving image jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1, 2, 3

[26] L. Zhu, Z. Zhou, and D. Hu. Globally consistent reconstruction of ripped-up documents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(1):1–13, 2008. 1