A Genetic Algorithm-Based Solver for Very Large Jigsaw Puzzles

Yufeng Jiang

1. Our proposed solution

Now remains the question of which piece of select from the available pieces bank and where to locate it in the child. A piece boundary is denoted by a pair (x_i, R), consisting of the piece number and a spatial relation. Given all existing boundaries, the operator checks whether there exists a piece boundary for which both parent agree on a piece x_j. If such a pixel exists, then it is placed in the correct location. If the parents agree on two or more boundaries, one of them is chosen at random and the respective piece is assigned. If there is no agreement between the parents on any piece at any boundary, the second phase begins. To understand this phase, first introduced by Pomeranz *et al.* [2] The pieces x_i and x_j are said to best-buddies if

$$\forall x_k \in Pieces, C(x_i, x_j, R_1) \geq C(x_i, x_k, R_1) \quad (1)$$
$$and$$
$$\forall x_p \in Pieces, C(x_j, x_i, R_2) \geq C(x_j, x_p, R_2) \quad (2)$$

where $Pieces$ is the set of all given image pieces and R_1 and R_2 are complementary spatial relations. In the second phase the operator checks whether one of the parents contains a piece x_j in spatial relation R of x_i which is also a best-buddy of x_i in that relation. If so, the piece is chosen and assigned. As before, if multiple best-buddy pieces are available, one is chosen at random. If a best-buddy piece is found but was already assigned, it is ignored and the search continues for other best-buddy pieces. Finally, if no best-buddy piece exists, the operator randomly selects a boundary and assigns it the most compatible piece available. To introduce mutation in the first and last phase the operator places, with low probability, an available piece at random, instead of the most compatible relevant piece available.

In summary, the operator uses repeatedly a three-phase procedure of piece selection and assignment, placing first agreed pieces, followed by best-buddy pieces and finally by the most compatible piece abailable. An assignment is only considered at relevant boundaries to maintain the contiguity of the kernel-growing image. The procedure returns to the first phase after every piece assignment due to the prospective creation of new boundaries.

# of Pieces	Avg. Best	Avg. Worst	Avg. Avg.	Avg. Std. Dev.
432	86.19%	80.56%	82.94%	2.62%
540	92.75%	90.57%	91.65%	0.65%
805	94.67%	92.79%	93.63%	0.62%
2360	85.73%	82.73%	84.62%	0.86%
3300	89.92%	65.42%	86.62%	7.19%
5015	94.78%	90.76%	92.04%	1.74%
10375	97.69%	96.08%	97.12%	0.45%
22834	92.02%	91.46%	91.74%	0.28%

Table 1. Results of running the GA 10 times on every image in every set under direct comparison; the best, worst, and average scores were recorded for every image.

1.1. Rationale

In a GA framework, good traits should be passed on to the child. Here, since position independence of pieces is encouraged, the trait of interest is captured by a piece's set of neighbors. Correct puzzle segments correspond to a correct placement of pieces next to each other. The notion that piece x_i is in spatial relation R to piece x_j is key to solving the jigsaw problem. Nevertheless, every chromosome accounts for a complete placement of all the pieces. Taking into account the random nature of the first generation, the procedure must actively seek the better traits of piece relations.

2. Experimental results

Cho *et al.* [1] introduced three measures to evaluate the correctness of an assembled puzzle, two of which were repeatedlly used in previous works. The direct method has been repeatedlly denounced [2] as being both less accurate and less meaningful due to its inability to cope with slightly shifted puzzle solutions. Figure 1 illustrates the drawbacks of the direct comparison and the superority of the neihbor comparison. Note that a piece arrangement scoring 100% according to one of the methods is, by definition, the full reconstruction of the original image and will also achieve a score of 100% when measured by the other method. Thus, unless stated otherwise, all results are under neighbor comparison. For the sake of completeness, our results under

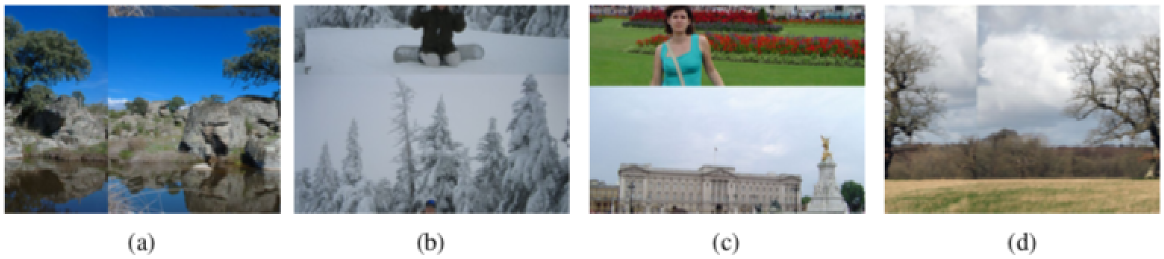

Figure 1. Shifted puzzle solutions. All images are solutions created by our GA. The accuracy for each solution is 0% according to the direct comparison, but over 95% according to the (more reasonable) neighbor comparison. Amazingly, the dissimilarity of each solution is smaller than that of its original image counterpart.

direct comparison are reported in Table 1.

In each generation we retain the best 4 chromosomes. The rest of the population is generated by the crossover operator described earlier with a mutation rate of 5%. Parent chromosomes are chosen by the roulette wheel selection, producing a single offspring in each crossover. The GA always runs for exactly 100 generations. We ran the proposed GA on the set of images supplied by Cho *et al.* [1] and all sets supplied by Pomeranz *et al.* [2], testing puzzles of 28 × 28-pixel patches according to the traditional convention.

3. Discussion and future work

In this paper authors presented an automatic jigsaw puzzle solver, far more accurate than any existing solver and capable of reconstructing puzzles of up to 22834 pieces. We also created new sets of large images to be used for benchmark testing for this and other solvers, and supplied both the image sets and our results for the benefit of the community [3].

Authors have achieved the first effective genetic algorithm-based solver, which appears to challenge state-of-the-art performance of other jigsaw puzzle solvers. By introducing a novel crossover technique, they were able to arrive at an effective solver.

References

[1] T. Cho, S. Avidan, and W. Freeman. A probabilistic image jigsaw puzzle solver. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 183–190, 2010. 1, 2

[2] 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*, pages 9–16, 2011. 1, 2

[3] D. Sholomon, O. David, and N. Netanyahu. Datasets of larger images and GA-based solvers results on these and other sets. http://www.cs.biu.ac.il/∼nathan/Jigsaw. 2