# Solving Jigsaw Puzzles using Paths and Cycles

Lajanugen Logeswaran
lajanugenl.14@cse.mrt.ac.lk

Department of Electronic and Telecommunication Engineering
University of Moratuwa
Sri Lanka

There has been a growing interest in image jigsaw puzzles with square shaped pieces. A solver takes as input square shaped patches of the same size belonging to an image and attempts to reconstruct the image. The key components of a jigsaw solver are a compatibility metric and an assembly algorithm. A compatibility metric uses the color content of the image patches to identify which pairs of pieces are likely to be neighbors in the correct assembly. More specifically, given puzzle pieces $x, y$ and a neighboring relationship $d \in \mathcal{D} = \{left, ight, top, bottom\} = \{l, r, t, b\}$ a compatibility metric $C$ assigns a numeric value $C(x, y, d)$ which represents how likely it is that piece $y$ is the neighbor of piece $x$ in the direction indicated by $d$. The assembly algorithm attempts to put the pieces together in the correct arrangement guided by these compatibility values. Prior work present several compatibility metrics and assembly algorithms.

We propose techniques which attempt to exploit more contextual information provided by the compatibilty metric compared to previous work. We introduce the concept of paths and cycles in jigsaw puzzles and show that they provide a means of identifying correct and incorrect matches. Based on this concept we propose refinement techniques which incrementally modify the compatibility values suggested by a metric to improve its neighbor identification accuracy. We further propose a means of exploiting information provided by different compatibility metrics. We define a compatibility measure based on the idea of cycles and use it to guide a greedy solver. The solver beats state of the art performance and the improvements are significant in the more challenging situation of smaller piece size. We briefly discuss the proposed techniques below.

A neighbor matrix $N$ represents point estimates $N(x, d)$ of the neighbor for each piece $x$ and direction $d$. Based on the raw compatibility scores we may obtain these estimates as $N(x, d) = \text{argmin}_y C(x, y, d)$. The piece identified as the best candidate to be the top neighbor of $x$ is $N(x, t)$. We may also observe that $N(N(N(x, l), t), r)$ is another estimate for the top neighbor of $x$. They may happen to be the same or different depending on the correctness of the entries in the neighbhor matrix (and whether $x$ is located in the left or top borders in the correct assembly). In general, we may consider a sequence of directions $\mathbf{d} = (d_1, d_2, ..., d_n)$ to obtain beliefs about piece placement $x_n$ at the location determined by $\mathbf{d}$, relative to a given piece $x$, where $x_n$ is defined as following : $x_0 = x, x_i = N(x_{i-1}, d_i) \forall i \in \{1..n\}$. We define the sequence of pieces $(x_0, x_1, .., x_n)$ to be a ***path***, and say that the *links* $(x_{i-1}, x_i, d_i)$ make up the path.

Consider the situation where the direction sequence $\mathbf{d}$ represents a closed curve (such as $(l, r), (l, t, r, b)$ etc.). For a path $(x_0, .., x_n)$ generated by such a direction sequence it has to be true that $x_n = x_0$ if all the links making up the path are correct. If not, we may conclude that atleast one of these links is incorrect. If the property does hold, intuitively this makes the constituent links likely to be correct. In this case we call the path a ***cycle***.

The idea of cycles motivated us to define an alternative measure of piece pair compatibility. We define the *strength* of a link $(x, y, d)$ to be the number of cycles to which it belongs. This link strength measure guides our proposed techniques for improving the neighbor identification accuracy of a given compatibility metric. The proposed **cost refinement** technique iteratively modifies the scores suggested by a compatibilty metric in an attempt to use correctly and confidently identified piece neighbors to correct piece neighbors identified incorrectly. The proposed **neighbor refinement** procedure makes use of paths starting and ending at the same two pieces to repair incorrect entries in a given neighbor matrix.

Different compatibility metrics may use different image features and techniques to score piece pairs. There is no single metric which performs best for all types of pieces and puzzles. Although one may be dominant when considering the overall performance we found that different metrics taken together have more to offer than the individual metrics. We thus propose a means of **combining the strengths of multiple compatibility metrics** using the cycles idea. The incremental improvements in neighbor identification accuracy contributed by each of the aforementioned techniques are illustrated for a particular puzzle in Figure 1.

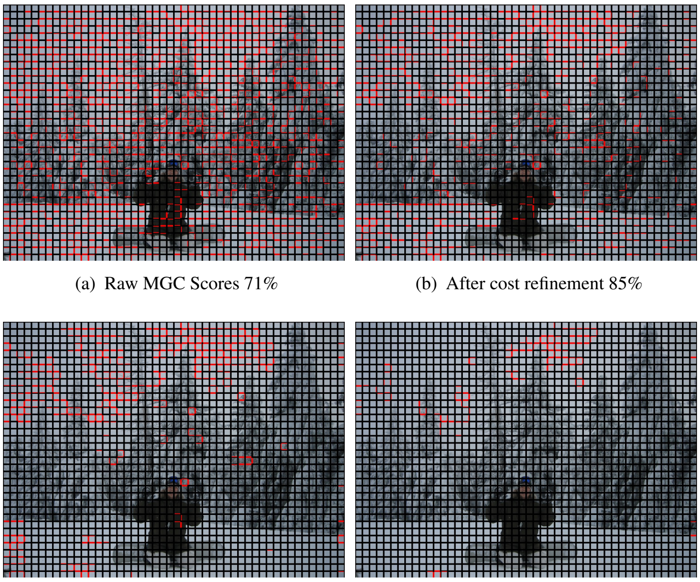
(a) Raw MGC Scores 71%
(b) After cost refinement 85%
(c) After neighbor refinement 91%
(d) After combining metrics 97%

Figure 1: Incorrect neighbor relationships (indicated by red markings on piece boundaries) after application of each improvement procedure - Image 11 of Cho et al.'s database [1] (piece size = 14, puzzle size = 1728)

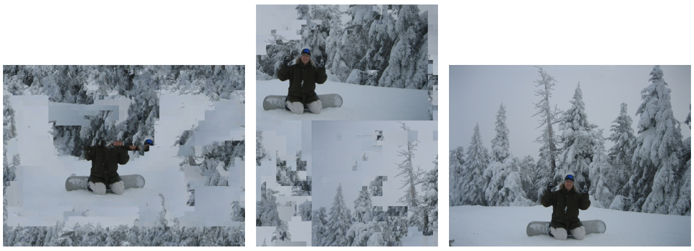
(a) Pomeranz et al. [3]
(b) Gallagher [2]
(c) Ours

Figure 2: Visual comparison of puzzle assembly with two recently proposed solvers (piece size = 14, puzzle size = 1728)

Although high neighbor identification accuracies are favorable, the quality of puzzle assembly depends equally well on the assembly algorithm. In a greedy approach the order in which piece pairs are picked is important. Early mistakes may adversely affect assembly, depending on the robustness of the algorithm. While previous work have used the compatibility scores directly either to determine the order to pick piece pairs in a greedy approach or to define an energy function which is optimized, we use our link strength measure to guide a greedy solver. Significant improvements are observed in puzzle assembly compared to previous work, especially in the more challenging case of smaller piece size. Figure 2 compares our assembly procedure with two previously proposed algorithms on a puzzle instance.

We plan to explore further ways in which paths and cycles may be utilized to build robust solvers in future, complementing the limitations of compatibility metrics in identifying correct neighbor relationships.

[1] Taeg Sang Cho, Shai Avidan, and William T. Freeman. A probabilistic image jigsaw puzzle solver. In *CVPR*, pages 183–190. IEEE, 2010.

[2] Andrew C. Gallagher. Jigsaw puzzles with pieces of unknown orientation. In *CVPR*, pages 382–389. IEEE, 2012.

[3] Dolev Pomeranz, Michal Shemesh, and Ohad Ben-Shahar. A fully automated greedy square jigsaw puzzle solver. In *CVPR*, pages 9–16. IEEE, 2011.