Solving Image Puzzles with A Simple Quadratic Programming Formulation

MEERJA MAQBUL BAIG[1], SHAHELA OSMANI[2]
[1] M-Tech, Dept. of computer science, Blogger and web developer Jyothishmathi Institute of Technology and Science.
[2] Sr Analyst -Support & SEO

Abstract- ***We present a new formulation to automatically solve jigsaw puzzles considering only the information contained on the image. Our formulation maps the problem of solving a jigsaw puzzle to the maximization of a constrained quadratic function that can be solved by a numerical method. The proposed method is deterministic and it can handle arbitrary rectangular pieces. We tested the validity of the method to solve problems up to 3300 puzzle pieces, and we compared our results to the current state-of-the-art, obtaining superior accuracy.***

I. INTRODUCTION

In the well-known jigsaw puzzle problem, numerous non overlapping tiles have to be assembled together, according to a color pattern or shape fitting, with the goal of reconstructing a single plane or image. Although this problem has been proven to be NP-complete when the affinity between the tiles is uncertain [1], several scientific challenges such as the reconstruction of documents from shredded paper [2], and reassembling broken archeological artifacts from fragments [3], can be reformulated as 2D or 3D jigsaw puzzle problems. In this paper we focus on the problem of reconstructing images from identically shaped rectangular tiles placed without repetition within a regular rectangular grid of known dimensions. Contrary to what occurs in traditional jigsaw puzzles, here the tile shape does not provide any information, making the problem even more challenging. In solving this kind of problem, we first need to deal with its combinatorial nature: since a tiling can be described as a permutation of the tiles within the rectangular grid, the number of possible tilings grows exponentially as a function of the number of tiles. In addition, since the problem is global in nature we seek local measures of pairwise tile matching to help reduce the complexity of the search.

However, no such rule based solely on local boundary tile similarity is known to date. Automatic solvers for jigsaw puzzles have been proposed since 1954, when the first method was presented by Freeman and Garder [4]. It was able to solve pictorial puzzles with 9 fragments by analyzing critical points on the border of the puzzle pieces. Since then and based on the first method, several other methods focused on matching the shape of the tiles only [5]. Kosiba et al. [6] were the first to consider not only the shape of the tiles, but also the content of the image. In their method, the matching process between the tiles considers many characteristics: color samples along the borders, curvature parameters, and the concavity and convexity of the tiles. Nielsen et al. [7] proposed the first solver to assemble successfully puzzles without shape information. The method was able to solve puzzles with 320 square tiles using a greedy approach.

Two recently proposed methods [8], [9] solve the square jigsaw puzzle by considering a pairwise compatibility metric between tiles which compares the color information on the shared boundary of two tiles. Cho et al. [8] presents a solver for puzzles with 432 tiles based on maximizing a probability function via loopy belief propagation. Since they don't have local evidence for their graphical model, they rely on knowing the placement of some tiles to solve the problem. Pomeranz et al. [9] presented the current state-of-the-art solver. By assuming a puzzle with square tiles, they solve problems with up to 3300 tiles using a greedy approach. However, this method requires solving each puzzle several times starting from different random seeds to obtain good results up to certain accuracy. In this paper we propose a simple

quadratic programming formulation to solve jigsaw puzzles with identically shaped rectangular tiles. We show that, for the same image sets, we can achieve superior accuracy over the current state-of-the-art method. Our method will be referenced as PSQP – Puzzle Solving by Quadratic Programming.

II. RELATED WORK

First consider an image partitioned into a regular 2D grid of size Ncols × Nrows, forming N tiles t1,...,tN , of identical dimensions. Now consider an empty grid of the same size as the previous one with N locations labeled 1,...,N. The problem is to determine a one-to-one correspondence between the N tiles and the N locations, optimal with respect to certain properly constructed global matching function. Since this correspondence can be described by a permutation π of the N tiles, the problem to be solved reduces to a discrete optimization problem over the finite group of permutations of N elements.

We organize the locations as a directed graph G = {V,E = EH ∪ EV }, where the vertices are the tile locations, V = {1,...,N}, and the set of edges E comprises all pairs of neighboring tile locations. EH and EV denote the set of horizontal and vertical neighboring locations, respectively. G must be a direct graph because in general, swapping two tiles from neighboring locations should result in a change in the global matching function. For each pair of tiles (ti, tj) so that $1 \leq i, j \leq N$ and i = j, we define two local matching compatibilities $CH_{i,j} \geq 0$ and $CV_{i,j} \geq 0$, that correspond to the compatibility of assigning ti and tj to locations connected by any horizontal edge $e \in EH$ or vertical edge $e \in EV$, respectively. We consider the following global matching function of a permutation π

where e = (i, j) is the edge connecting the neighboring locations i and j, and π(i) can be regarded as a 1-1 mapping which assigns the tile tπ(i) to the location i. Our goal is to maximize this function over all the permutations π of N elements. Since this is a hard combinatorial optimization problem, we first extend the domain of the global matching function to the set of doubly stochastic matrices, and we reformulate the problem as a constrained continuous optimization problem, which we solve using numerical methods. Then we describe how the coefficients of the matrices CH and CV are computed so that the solution of the continuous optimization problem correlates well with the original combinatorial optimization problem.

2.1 Global Matching Function

In this section we show how Equation 1 can be reformulated as a homogeneous quadratic function of a square matrix, which allows us to relate the problem to a simpler continuous optimization problem. First of all, each permutation π of N elements can be represented as a permutation matrix, i.e., a binary square matrix P with exactly one entry equal to 1 in each row and in each column:

With this notation we can reformulate the global function as follows

where a generic term (P CP)ij , corresponding to the edge e = (i, j), is the element (ij) of the square matrix (P CP). Note that for each edge e = (i, j), the term (P CP) is a homogeneous non-negative quadratic function of elements of matrix P. It follows that the sum of all terms of ε(P) is also a homogeneous non-negative quadratic function of P. If we represent the columns of the N × N matrix P as a vector p of dimension N2, we get where p is the vertical concatenation of the columns, p1,...,pN of P. We can reformulate Equation 4 in the canonical form pAp, where A is a symmetric N2 × N2 matrix, representing the Hessian of ε(P). In vector form and in coordinates:

Even though in practice we never construct the matrix A explicitly, an analytic expression can be obtained. Since for a generic matrix C we can write and it follows that the coefficients of the matrix A can be accumulated by a simple linear traversal of matrices CH and CV . Fig. 1 illustrates the problem formulation.

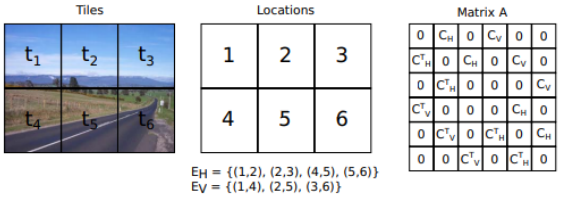


Fig. 1. Problem formulation. From left to right: tiles, locations with the two edge sets, and matrix A

represented as a block matrix. C H and C V are the transpose of matrices CH and CV , respectively, and each block of matrix A is a 6 × 6 matrix.

III. PROPOSED METHOD

3.1 Constrained Gradient Ascent

Permutation matrices are special cases of doubly stochastic matrices [10]. A doubly stochastic matrix is a non-negative matrix such that the sum of all the elements in each row is equal to 1, and the sum of all the elements in each column is also equal to 1. In fact, the set of doubly stochastic matrices is the convex hull of the permutation matrices within the set of N × N matrices. Each doubly stochastic matrix satisfies N2 inequality constraints, which specify that the elements Pij of P are non-negative; and 2N equality constraints, which specify that the sum of the rows and columns of P are equal to 1. By extending the domain of ε(P) to all the 64 doubly stochastic matrices, the problem reduces to solving the following quadratic optimization problem where there is a column vector of size N with all elements equal to one. We use a constrained gradient ascent algorithm, with gradient projection [11], to search for local maxima of this problem. Note that even though the objective function f(p) is positive on the feasible set, it is not necessarily concave because matrix A is not positive definite: all the diagonal values of A are 0 in our formulation, which violates the necessary conditions for positive definiteness, and also for positive semi-definiteness. Therefore, we cannot guarantee f(p) to attain a maximum at a permutation matrix. But in practice, we observe that we can get as close as possible to a solution by working with the constraints. To maximize Equation 8, we propose a modified constrained gradient ascent approach, with gradient projection [11]. To locate a local maxima of a function, we need to update the variables in steps proportional to the gradient at the current point, while projecting the gradient. In our approach we maintain a set of active variables. An inactive variable is one that is at the boundary of the feasible region and cannot be further updated. The method starts from pkl = 1 N , 1 ≤ k,l ≤ N, and with all variables active, active kl = true, where active indicates if a variable is active or not. The ascent direction, d = ∇f(p) = A ∗ p, at the current estimate p, in general does not satisfy the linear constraints, so we must project it onto the space orthogonal to the subspace defined by the linear equality constraints [11], resulting in the constrained ascent direction c. Fig. 2 illustrates a 2D simplification of the process of projecting the grad

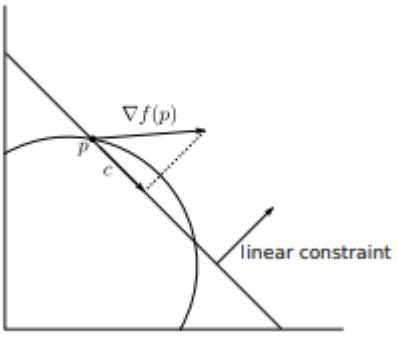


Fig. 2. Projection of the gradient onto the space orthogonal to the space defined by the linear equality constraints. With the constrained ascent direction c, the method can update the previously feasible point to a new feasible point p: pkl = pkl + step ∗ c, for 1 ≤ k,l ≤ N, and active kl = true, where step is the maximum value so that 0 ≤ pkl ≤ 1

When one of the variables reaches the boundary of the feasible region, we should update the constraints so that this variable stays at the boundary. However, in practice, maintaining a group of modifying and orthogonal constraints implies high computational costs and storage. Instead, we reinitialize p every time there is no direction to maximize the energy inside the feasible region. In order to do this, we deactivate the variables that are on the limit of the feasible region, i.e., the ones that are equal to either zero or one. The process of deactivating a variable on the upper limit corresponds to assigning the corresponding tile to the most probable location. We then restart p without the inactive variables, i.e., pkl = 1 N − nFixedTiles , for 1 ≤ k,l ≤ N, and active = true, where nFixedTiles is the number of tiles that have been assigned to a location. Then we repeat these steps until all the tiles have been assigned to a location. Algorithm 1 shows the pseudo-code for the ascent gradient approach that maximizes Equation

3.2 Compatibility between tiles

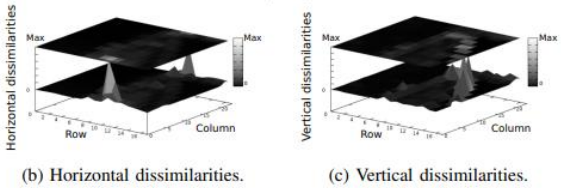
The compatibility between pairs of tiles has been studied before [8], [9], and plays an important role in

solving the image puzzle. Demaine et al. [1] showed that if it is locally possible to tell whether two tiles fit together in the final solution, then trying to join together all pairs of tiles in a greedy manner solves the puzzle in polynomial time. But, in natural images, it is easy to find examples of tiles with ambiguous neighboring tiles. In this work, we consider the prediction based compatibility proposed by Pomeranz et al. [9]. The horizontal dissimilarity between a left hand side tile right hand side tile tj is defined as

where tiles ti and tj are regarded as T × T × 3 matrices, variables p and q are tunable parameters, and the color difference is measured in the normalized LAB color space. The vertical matching error DVij is computed in a similar fashion. Based on the predicted values, the compatibility between tiles ti and tj is defined as [9]

where quartile(i) is the quartile of the dissimilarity among all other tiles and tile ti. Unfortunately, the local matching dissimilarity (Equation 9) does not provide enough information to solve the puzzle globally. This is illustrated in Fig. 3, where it is shown the dissimilarities considering only correctly assigned neighboring tiles. We can see that in some constant parts of the image (the sky, for example), the dissimilarity among all tiles is lower than in other non-constant parts and thus they are not comparable. The problem gets worse when we consider the errors between every possible pair of tiles.

(a) Correctly assigned tiles.



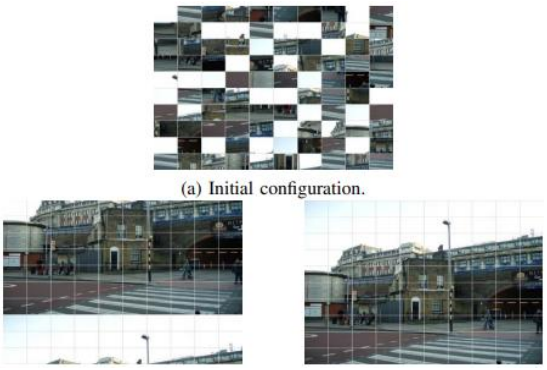
(b) Horizontal dissimilarities. (c) Vertical dissimilarities.

Fig. 3. Dissimilarity between correctly assigned tiles.

Due to this difficulty, we present a new compatibility measure, based on [9], that imposes a stronger global order to the tiles' dissimilarities. The horizontal compatibility between tiles ti and tj is defined as

3.3 Implementation

In the implementation of the gradient ascent method, the memory footprint is a major concern, because matrices CH and CV , vector p, and the ascent direction c have N × N entries each. To save up memory, we observed that the term $\phi(i)$ (Equation 11) makes the compatibility values really small when distant neighbors of it are considered. Thus, using a safe threshold (10^{-6}) we can zero out compatibility values that are already almost zero. By doing this, matrices CH and CV become sparse. In the optimal case, CH will have Nrows(Ncols − 1) non-zero entries and CV will have Ncols(Nrows − 1) non-zero entries, a drastic reduction in memory usage. In terms of computational complexity, our algorithm runs in quadratic time in the number of tiles, i.e, PSQP is O(n2). The ascent direction computation is done by traversing matrices CH and CV , and the projection of the ascent direction is done by traversing the corresponding vector two times. In practice, we have two problems that were not discussed before. First, the constant tiles – groups of tiles that have equal feature vectors on all sides – impose a hard problem to solve. These tiles have total compatibility among them and to the neighboring non-constant tiles. To address this, we simply do not take them into account by zeroing out their compatibility. By doing this, the constant tiles will fit in the holes left by the optimization process. Note that, because they are equal, it doesn't matter which permutation will be adopted among them. The second problem is the non-convexity property of the energy function that we want to maximize. There is no guarantee that the maximum provided by the Constrained Gradient Ascent algorithm is the global maximum, only a local one. For a few puzzles, especially the ones that contain constant tiles (and the compatibility values were altered), the final permutation does not represent a global maximum, but a local maximum that is a shift of the global permutation (Fig. 4).



(a) Initial configuration.

(b) Permutation associated with a local maximum.
(c) Permutation associated with the global maxima.

Fig. 4. Example of the non-concavity property of the global matching function. Note that this puzzle contains several constant (white) tiles.

IV. RESULTS

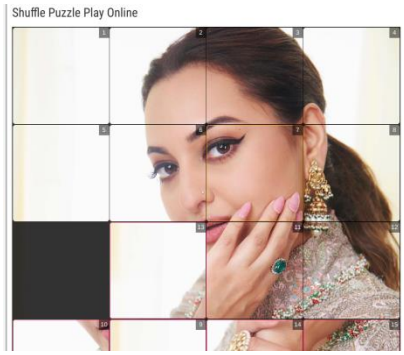


Fig.1 Image for medium option in the shuffle puzzle play online

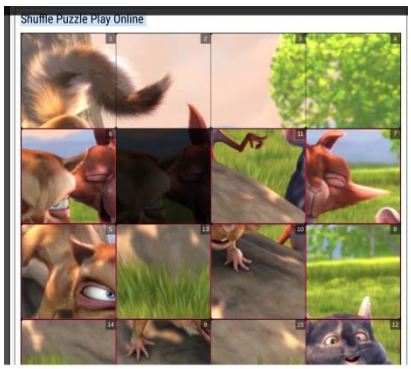


Fig.2 Image for "Hard" option in the shuffle puzzle play online

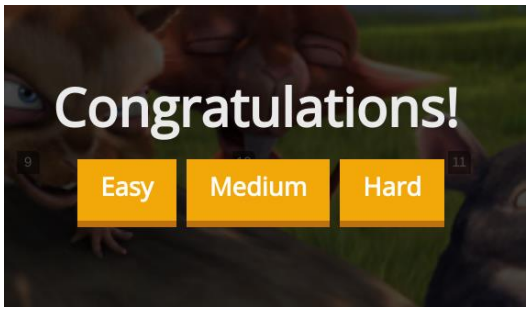


Fig.3 Results after the successful implementation of the puzzle

CONCLUSION

We introduced a new formulation for solving image jigsaw puzzle problems, the method PSQP – Puzzle Solving by Quadratic Programming. In our formulation, a solved puzzle is a one-to-one assignment of tiles to locations, according to an energy function. Since this is a hard combinatorial problem, we reformulate it as a quadratic programming approach, where we can find an approximate solution by means of a gradient ascent algorithm. We compared PSQP to the current state-of-the-art and it provided superior results according to the used metrics. PSQP also has some advantages. First, it can solve puzzles not only with square tiles, but also with rectangular ones. Second, it is deterministic and although several parameter sets have to be tested, the method always yields the same results, while the current state-of-the-art method has to be executed several times to attain a certain accuracy. For the size of the puzzles tested, PSQP is faster, considering all the necessary executions in both methods. By analyzing the results, we observed that image puzzles that contain constant tiles are a weakness of PSQP. Constant tiles are difficult to order in a global sense, so we cannot consider them as a normal piece. We also observed that the right parameter set for each image may be determined a priori by analyzing the image and tiles properties. These two observations will be included in future studies.

REFERENCES

[1] E. Demaine and M. Demaine, "Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity," Graphs and Combinatorics, vol. 23, pp. 195–208, 2007.

[2] E. Justino, L. Oliveira, and C. Freitas, "Reconstructing shredded documents through feature matching," Forensic science international, vol. 160, no. 2, pp. 140–147, 2006.

[3] J. McBride and B. Kimia, "Archaeological fragment reconstruction using curve-matching," in Conference on Computer Vision and Pattern Recognition Workshop. (CVPRW), vol. 1, 2003, pp. 3–3.

[4] H. Freeman and L. Garder, "Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition," IEEE Transactions on Electronic Computers, no. 2, pp. 118–127, 1964.

[5] D. Goldberg, C. Malon, and M. Bern, "A global approach to automatic solution of jigsaw puzzles," in Proceedings of the eighteenth annual symposium on Computational geometry, 2002, pp. 82–87.

[6] D. Kosiba, P. Devaux, S. Balasubramanian, T. Gandhi, and K. Kasturi, "An automatic jigsaw puzzle solver," in Proceedings of the 12th International Conference on Pattern Recognition (IAPR), vol. 1, 1994, pp. 616–618.

[7] T. Nielsen, P. Drewsen, and K. Hansen, "Solving jigsaw puzzles using image features," Pattern Recognition Letters, vol. 29, no. 14, pp. 1924–1933, 2008.

[8] T. Cho, S. Avidan, and W. Freeman, "A probabilistic image jigsaw puzzle solver," in Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 183–190.

[9] 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, pp. 9–16.

[10] E. Seneta, Non-negative matrices and Markov chains. Springer Verlag, 2006.

[11] J. Rosen, "The gradient projection method for nonlinear programming. part i. linear constraints," Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 1, pp. 181–217, 1960.

[12] https://dailypuzzlecheats.com/

[13] https://dailypuzzlecheats.com/shuffle-puzzle-play-online