ON THE SECURITY OF BLOCK SCRAMBLING-BASED ETC SYSTEMS AGAINST JIGSAW PUZZLE SOLVER ATTACKS

Tatsuya CHUMAN, Kenta KURIHARA, and Hitoshi KIYA

Tokyo Metropolitan University Tokyo, 191-0065, Japan

ABSTRACT

This paper is among the first to adapt automatic jigsaw puzzle solvers, which are methods of assembling jigsaw puzzles, to the field of information security. Block-based scrambling encryption schemes, which have been mainly proposed for Encryption-then-Compression (ETC) systems, have enough key spaces for protecting brute-force attacks. However, each block in encrypted images has almost the same correlation as that of original images. Therefore, it is required to consider the security from different viewpoints from number theory-based encryption methods with provable security such as RSA and DES. In this paper, existing jigsaw puzzle solvers are first reviewed in terms of attacking strategies on encrypted images. Then, a new jigsaw puzzle solver is proposed to extend some limitations of the conventional ones. In the experiments, the jigsaw puzzle solvers are applied to encrypted images to consider the security conditions of the encryption schemes.

Index Terms— jigsaw puzzle, JPEG, encryption, ETC system

1. INTRODUCTION

The use of images and video sequences has greatly increased recently because of the rapid growth of the Internet and multimedia systems. A lot of studies on secure, efficient and flexible communications have been reported [1–3]. For securing multimedia data, full encryption with provable security (like RSA, AES, etc) is the most secure option. However, many multimedia applications have been seeking a trade-off in security to enable other requirements, e.g., low processing demands, retaining bitstream compliance, and signal processing in the encrypted domain, so that a lot of perceptual encryption schemes have been studied as one of the schemes for achieving the trade-off. They can also be combined with the encryption methods with provable security.

In this paper, we focus on block scrambling-based image encryption schemes, which have been proposed for Encryption-then-Compression (ETC) systems with the assumption of international compression standards to consider the safety [4–8]. So far, the safety has been evaluated based on its key space assuming the brute-force attacks, so that the schemes have enough key spaces for protecting the attacks. However, each block in encrypted images has almost the same correlation as that of original images. Several efficient attacks on the permutation-only encryption have been studied [9, 10], but they are not available for the block scrambling-based encryption.

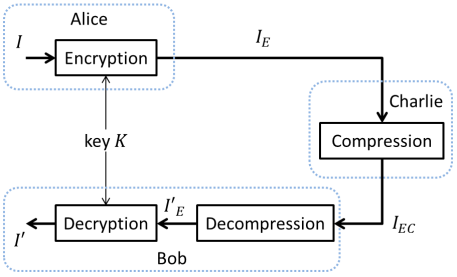


Fig. 1. Encryption-then-Compression system

On the other hand, recently, jigsaw puzzle solvers, that utilize the correlation between pieces, have succeeded in solving puzzles with a large number of pieces [11–20]. In this paper, we regard the blocks of an encrypted image as pieces of a jigsaw puzzle and evaluate the safety of the encryption assuming the jigsaw puzzle solvers as crypto-attack methods. Existing jigsaw puzzle solvers are first reviewed in terms of attacking strategies on encrypted images. Then we point out to need new types of jigsaw puzzle solvers for the attacks, and propose a new solver to extend some limitations of conventional ones.

Finally, we evaluate the safety of the encryption by applying the jigsaw puzzle solvers to encrypted images. It is shown that some solvers can decrypt encrypted images even when the key space is large enough. On the other hand, it is also confirmed that an appropriate selection of the block size and the encryption methods makes the decryption of images difficult.

2. PREPARATION

2.1. Block scrambling-based image encryption

Block scrambling-based image encryption schemes have been proposed for ETC systems [5–8], in which a content owner, Alice, wants to securely transmit an image I to a recipient, Bob, via an untrusted channel provider, Charlie, as

This research was (partly) supported by Grant-in-Aid for Research on Priority Areas, Tokyo Metropolitan University, Research on social big data.

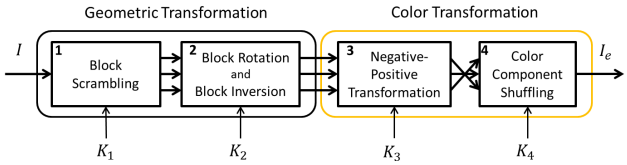
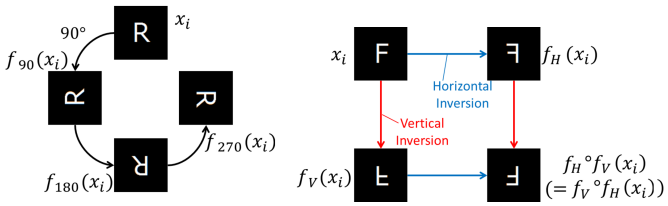


Fig. 2. Block scrambling-based image encryption



(a) Block rotation (b) Block inversion

Fig. 3. Block rotation and inversion

illustrated in Fig. 1.

In the schemes [4–8], an image with $X \times Y$ pixels is first divided into non-overlapped blocks with $B_x \times B_y$, then four block-based processing steps, as illustrated in Fig. 2, is applied to the divided image. The procedure of performing the image encryption to generate an encrypted image I_e is given as follows:

Step1: Divide an image with $X \times Y$ pixels into blocks with $B_x \times B_y$ pixels, and permute randomly the divided blocks using a random integer generated by a secret key K_1, where K_1 is commonly used for all color components.

Step2: Rotate and invert randomly each block (see Fig. 3) using a random integer generated by a key K_2, where K_2 is commonly used for all color components as well.

Step3: Apply the negative-positive transformation to each block using a random binary integer generated by a key K_3, where K_3 is commonly used for all color components. In this step, a transformed pixel value in ith block B_i, p' is computed by

$$p' = \begin{cases} p & (r(i) = 0) \\ p \oplus (2^L - 1) & (r(i) = 1) \end{cases} \quad (1)$$

where $r(i)$ is a random binary integer generated by K_3 and $p \in B_i$ is the pixel value of an original image with L bpp.

Step4: Shuffle three color components in each block (the color component shuffling) using a random senary integer generated by a key K_4.

2.2. Key space analysis

If an image with $X \times Y$ pixels is divided into blocks with $B_x \times B_y$ pixels, the number of blocks n is given by

$$n = \lfloor \frac{X}{B_x} \rfloor \times \lfloor \frac{Y}{B_y} \rfloor \quad (2)$$

where $\lfloor \cdot \rfloor$ is the function that rounds down to the nearest integer.

Table 1. A summary of latest square jigsaw puzzle solvers. "○" indicates support for puzzles with unknown rotation or inversion. "×" indicates unsupport for puzzles with unknown rotation or inversion.

Methods	Authors	Rotation	Inversion	Year	Pieces
Greedy	Pomeranz [11]	×	×	2011	3300
	Gallagher [12]	○	×	2012	9600
	Mondal [13]	○	×	2013	540
	Son [14]	○	×	2014	9801
	Son [15]	○	×	2016	3300
Global	Cho [16]	×	×	2010	432
	Andalo [17]	×	×	2012	3300
	Sholomon [18]	○	×	2014	22755
	Sholomon [19]	×	×	2016	30745
Hybrid	Rui [20]	○	×	2015	3300

The key space of the block scrambling (Step1) N_S, which is the number of permutation of n blocks, is given by

$$N_S = {}_nP_n = n!. \quad (3)$$

Similarly, the key spaces of other encryption steps are given as

$$N_R = 8^n,\ N_N = 2^n,\ N_C = \left({}_3P_3\right)^n = 6^n \quad (4)$$

where N_R, N_N and N_C are the key spaces of the encryption combining the block rotation and the block inversion (Step2), the negative-positive transformation (Step3) and the color component shuffling (Step4) respectively. Consequently, the key space of encrypted images by using all the proposed encryption steps, N_A, is represented by

$$\begin{aligned} N_A &= N_S \cdot N_R \cdot N_N \cdot N_C \\ &= n! \cdot 8^n \cdot 2^n \cdot 6^n. \end{aligned} \quad (5)$$

The key space is expanded by combining some independent encryption steps. As a result, when an encrypted image has more than $n = 28$ blocks, the key space of the image is larger than that of the 256-bit key. Thus, the key space of the scheme is generally large enough against the brute-force attacks.

However, an encrypted image has almost the same correlation among pixels in each block as that of the original image, whose property enables to efficiently compress images. Therefore, an attacker can utilize the correlation to decrypt the image in some way. The aim of this paper is to discuss the security of the encryption against jigsaw puzzle solver attacks that are based on the correlation.

3. JIGSAW PUZZLE SOLVER ATTACKS

Jigsaw puzzle solver is a method of assembling jigsaw puzzles. In the block scrambling-based encryption, if we regard the blocks as pieces of a jigsaw puzzle, decrypting encrypted images is similar to assembling the jigsaw puzzle. Therefore, jigsaw puzzle solvers are considered as one of the attack methods on the block scrambling-based encryption in this paper.

3.1. Related works

Jigsaw puzzle solvers are broadly classified into three categories according to their assembly strategies, i.e., greedy

Table 2. Jigsaw puzzle types

Type	Scramble	Rotation	Inversion	Negative-Positive Transformation
Type 1	✓			
Type 2	✓	✓		
Type I	✓	✓	✓	
Type N	✓	✓		✓
Type IN	✓	✓	✓	✓

methods, global methods and their hybrid methods [20]. The greedy methods start from initial pairwise matches and successfully build larger and larger components. On the other hand, the global methods directly search for a solution by maximizing a global compatibility function. Table 1 shows typical solvers. For example, the jigsaw puzzle solver [19] completely succeeded in assembling large puzzles with 30745 pieces in 2016.

On the other hand, a solver for puzzles including rotated pieces (pieces with unknown orientation) was first proposed in 2012 [12]. Thus, even when the number of blocks in an encrypted image is larger than 30000, there is a possibility that the image is completely decrypted. In this paper, jigsaw puzzle solvers are among the first to be considered as one of attacks on the image encryption.

The existing jigsaw solvers do not support inverted or negative-positive transformed pieces. Therefore, we define new types of jigsaw puzzles, as shown in Table 2, where Type I, Type N and Type IN are new. Examples of encrypted images are illustrated in Fig. 4(b) and (c), where Fig. 4(a) is the original one. As shown in Fig. 4(e) and Fig. 4(f), recognizing objects in two assembled images is difficult unlike Fig. 4(d).

3.2. Jigsaw puzzle solver for new type puzzles

The greedy method [12] is extended as a new jigsaw puzzles solver to assemble puzzles including inverted piece or negative-positive transformed piece. The following is the procedure.

3.2.1. Pairwise compatibility

To calculate pairwise compatibility between pieces, we use Mahalanobis Gradient Compatibility (MGC) proposed by Gallagher [12]. Given the pieces x_i and x_j, $i, j = 1, 2, \ldots, n$, the compatibility between the right side of x_i and the left side of x_j is expressed as $C_{LR}(x_i, x_j)$.

3.2.2. Pairwise comparison

We represent transform function that rotates x_j 0°, 90°, 180° or 270° as f_R, $R \in \{0, 90, 180, 270\}$ shown in Fig. 3(a). The function that inverts x_j horizontally(H) or vertically(V) is defined as $f_I(x_j)$, $I \in \{H, V, 0\}$ as in Fig. 3(b), where $f_0(x_j)$ is the fucnction that indicates non-inverted. $f_N(x_j)$, $N \in \{N, 0\}$ is the function whether applies negatve-positive transformation(N) to x_j. In addition to three transform functions, i.e., $f_R(x_j)$, $f_I(x_j)$, $f_N(x_j)$, the combination of them gives other transformations. Then, a rotated, inverted, negative-positive transformed piece is represented as

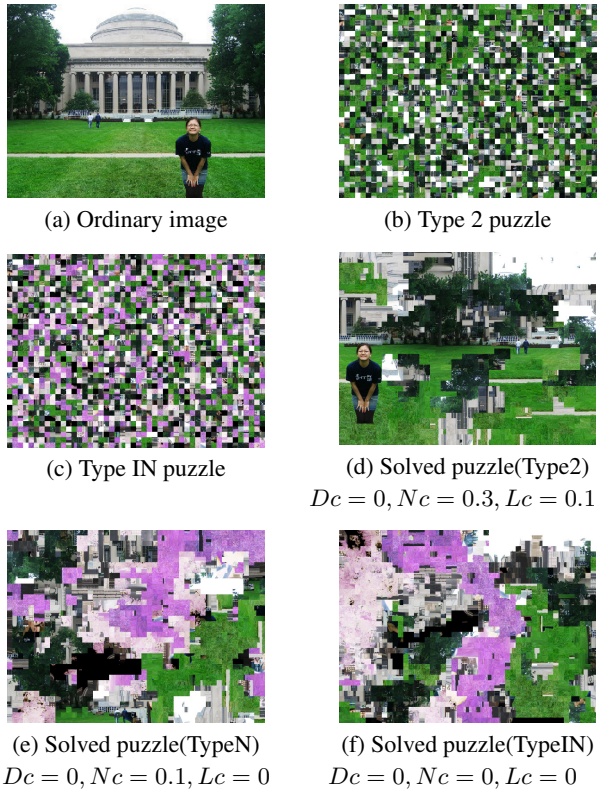
$$f_{R,I,N}(x_j) = f_R \circ f_I \circ f_N(x_j) \tag{6}$$

where $f_{R,I,N}(x_j)$ is the composite function which consists of three transform functions.

In the proposed solver, the minimum compatibility between the right side of x_i and the left side of x_j is defined by

$$\min C_{LR}(x_i, x_j) = \min_{f_{R,I,N}} \{C_{LR}(x_i, f_{R,I,N}(x_j))\}. \tag{7}$$

Finally, these minimum compatibility values are used to assemble jigsaw puzzle by using tree-based assembly method [12].



(a) Ordinary image (b) Type 2 puzzle (c) Type IN puzzle (d) Solved puzzle(Type2) $Dc = 0, Nc = 0.3, Lc = 0.1$ (e) Solved puzzle(TypeN) $Dc = 0, Nc = 0.1, Lc = 0$ (f) Solved puzzle(TypeIN) $Dc = 0, Nc = 0, Lc = 0$

Fig. 4. Examples of encrypted images and assembled images ($n = 1728$, $B_x \times B_y = 14$)

4. EXPERIMENTS AND RESULTS

4.1. Experimental conditions

Assembled image I_d from Type N or Type IN puzzle was compared with the original image I. The following three measures [12] [16] were used to evaluate the results.

Direct comparison (Dc): represents the ratio of the number of pieces which are in the correct position. Dc for image I_d, namely, $Dc(I_d)$ is calculated as

$$Dc(I_d) = \frac{1}{n} \sum_{i=1}^{n} d_c(i),$$

$$d_c(i) = \begin{cases} 1, \text{ if } I_d(i) \text{ is in the correct position} \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

Table 3. Evaluation of existing solver [12](n=432)

Piece Size	28×28 pixels			14×14 pixels		
Encryption Types	Type 2	Type N	Type IN	Type 2	Type N	Type IN
$Dc(I_d)$(Average)	0.822	0.022	0.012	0.377	0.010	0.013
$Nc(I_d)$(Average)	0.904	0.126	0.061	0.626	0.082	0.045
$Lc(I_d)$(Average)	0.889	0.109	0.048	0.551	0.055	0.033

Table 4. Evaluation of extended solver (n=432)

Piece Size	28×28 pixels		14×14 pixels	
Encryption Types	Type N	Type IN	Type N	Type IN
$Dc(I_d)$(Average)	0.780	0.394	0.252	0.010
$Nc(I_d)$(Average)	0.786	0.554	0.420	0.131
$Lc(I_d)$(Average)	0.795	0.572	0.415	0.126

where $I_d(i)$ represents the position of a piece i in image I_d

Neighbor comparison (Nc): is the ratio of the number of correctly joined blocks. Nc for image I_d, namely, $Nc(I_d)$ is calculated as

$$Nc(I_d) = \frac{1}{B}\sum_{k=1}^{B} n_c(k),$$
$$n_c(k) = \begin{cases} 1, \text{ if } b_k \text{ is joined correctly} \\ 0, \text{ otherwise} \end{cases} \quad (9)$$

where B is the number of boundaries among pieces in I_d, and b_k is the kth boundary in I_d. For an image with $u \times v$ blocks, there are $B = 2uv - u - v$ boundaries in the image.

Largest Component (Lc): is the ratio of the number of the largest joined blocks which have correct adjacencies to the number of blocks in an image. Lc for image I_d, namely, $Lc(I_d)$ is calculated as

$$Lc(I_d) = \frac{1}{n}\max_j\{l_c(I_d, j)\}, j = 1, 2, \cdots, m \quad (10)$$

where $l_c(I_d, j)$ is the number of blocks in the jth partial correctly assembled area, and m is the number of partial correctly assembled areas.

In the measures, $Dc(I_d), Nc(I_d), Lc(I_d) \in [0, 1]$, a larger value means a higher compatibility.

We used 20 images from MIT dataset, provided by Cho [16]. Five different encrypted images were generated by random keys from one ordinary image for each Type puzzle. We assembled the encrypted images by using jigsaw puzzle solvers and chose the image which had the highest sum of $Dc(I_d)$, $Nc(I_d)$ and $Lc(I_d)$ in those of five images. We performed these procedures for each type puzzle independently, and the average of 20 images was calculated for $Dc(I_d)$, $Nc(I_d)$ and $Lc(I_d)$.

4.2. Experiment result

A. Compression performance of the ETC system

Fig. 5 shows the Rate-Distortion (RD) curves of JPEG compressed images without any encryption and with the block scrambling-based image encryption [5, 6], where the average bitrate and PSNR values of 20 images were plotted, after decrypting the images. It is certified that the compression efficiency of the encrypted images is approximately equivalent to that of the original images. Therefore, it is certified that images encrypted by the block scrambling-based image encryption are not affected by the JPEG compression.

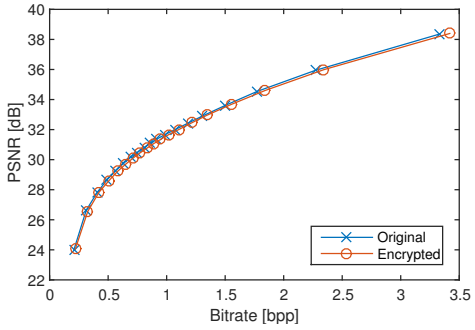


Fig. 5. RD curves of original images and encrypted ones ($X \times Y = 672 \times 504, B_x \times B_y = 16 \times 16$)

B. Existing jigsaw puzzles solver [12]

Table 3 shows the scores of images assembled by the existing jigsaw puzzle solver [12]. It is shown that the use of encrypted images with small size of blocks makes assembling the images difficult. Thereby, it is important to select the appropriate block size for the security of the encryption. Also, the assembly of encrypted images including inverted, negative-positive transformed or color component shuffled pieces is more difficult than that of Type 2 puzzles because, the existing solvers do not support these pieces.

C. Extended jigsaw puzzles solver

Table 4 summarizes the scores of the extended jigsaw puzzles solvers discussed in 3.2. As shown in Table 4, Type N puzzles ($B_x \times B_y = 28 \times 28$) were assembled about 80 percent by using the extended jigsaw puzzle solver. However, in the case of Type IN puzzles, the scores become much lower than other types. It is also confirmed that the scores of the solvers strongly depend on the size of pieces. Type IN puzzles ($B_x \times B_y = 14 \times 14$) could be assembled only 10 percent. Moreover, we evaluated the score of other type puzzles including the color component shuffling. It was confirmed that combining the encryption steps makes puzzle solvers more difficult than single use of each step.

5. CONCLUSION

In this paper, the safety of the block-scrambling based image encryption schemes was discussed. We focused on jigsaw puzzle solvers as one of attack methods on the encryption, and regarded blocks of an encrypted image as pieces of a jigsaw puzzle, although the safety has been evaluated so far on the size of the key space, assuming the brute-force attacks. Moreover, a existing jigsaw puzzle solver was extended to be adapted to the encryption schemes, and some jigsaw puzzle solvers including the proposed one were applied to encrypted images. In the simulations, the studies presented evidence that the appropriate selection of the block size and the combination of each encryption step can improve the strength of ETC against jigsaw puzzle solver attacks.

6. REFERENCES

[1] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varadharajan, and C-C. J. Kuo, "Survey on securing data storage in the cloud," *APSIPA Transactions on Signal and Information Processing*, vol. 3, e7, 2014.

[2] R. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, 2013.

[3] J. Zhou, X. Liu, O. C. Au, and Y. Y. Tang, "Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation," *IEEE transactions on information forensics and security*, vol. 9, no. 1, pp. 39–50, 2014.

[4] O. Watanabe, A. Uchida, T. Fukuhara, and H. Kiya, "An encryption-then-compression system for jpeg 2000 standard," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1226–1230.

[5] K. Kurihara, S. Shiota, and H. Kiya, "An encryption-then-compression system for jpeg standard," in *Proceedings of Picture Coding Symposium (PCS)*. IEEE, 2015, pp. 119–123.

[6] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for jpeg/motion jpeg standard," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 98, no. 11, pp. 2238–2245, 2015.

[7] K. Kurihara, O. Watanabe, and H. Kiya, "An encryption-then-compression system for jpeg xr standard," in *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2016, pp. 1–5.

[8] K. Kurihara, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for lossless image compression standards," *IEEE transactions on information and systems*, vol. E100-D, no. 1, pp. 52–56, 2017.

[9] C. Li and K. T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal processing*, vol. 91, no. 4, pp. 949–954, 2011.

[10] A. Jolfaei, X. W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 235–246, 2016.

[11] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, "A fully automated greedy square jigsaw puzzle solver," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[12] A. Gallagher, "Jigsaw puzzles with pieces of unknown orientation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] D. Mondal, Y. Wang, and S. Durocher, "Robust solvers for square jigsaw puzzles," in *Proceedings of the Conference on Computer and Robot Vision (CRV)*, 2013, pp. 249–256.

[14] K. Son, J. Hays, and D. B. Cooper, "Solving square jigsaw puzzles with loop constraints," in *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, 2014, pp. 32–46.

[15] K. Son, D. Moreno, J. Hays, and D. B. Cooper, "Solving small-piece jigsaw puzzles by growing consensus," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] T. Cho, S. Avidan, and W. Freeman, "A probabilistic image jigsaw puzzle solver," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[17] F. A. Andal, G. Taubin, and S. Goldenstein, "Solving image puzzles with a simple quadratic programming formulation," in *Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2012.

[18] D. Sholomon, O. E. David, and N. S. Netanyahu, "A generalized genetic algorithm-based solver for very large jigsaw puzzles of complex types," in *National Conference on Artificial Intelligence (AAAI)*, 2014.

[19] D. Sholomon, O. E. David, and N. S. Netanyahu, "An automatic solver for very large jigsaw puzzles using genetic algorithms," *Genetic Programming and Evolvable Machines*, vol. 17, no. 3, pp. 291–313, 2016.

[20] R. Yu, C. Russell, and L. Agapito, "Solving jigsaw puzzles with linear programming," *arXiv preprint arXiv:1511.04472*, 2015.