# Reconstructing 2D Art with Genetic Algorithms and Deep Learning

ELIZABETH STEVENS

UMM CSCI SENIOR SEMINAR

16 NOV 2019

https://artsandculture.google.com/partner/national-azulejo-museum

# Wall Paintings

https://www.ancient-origins.net/artifacts-other-artifacts/ladies-blue-0011518


# Tile Panels

https://artsandculture.google.com/partner/national-azulejo-museum

# Why does this matter?

- There are many fragmented mosaics, murals, frescoes, pottery, shredded documents, etc.
- Object reconstruction is hard, time consuming process
- Over a 100,000 tile panels stored at the National Tile Museum in Lisbon, Portugal

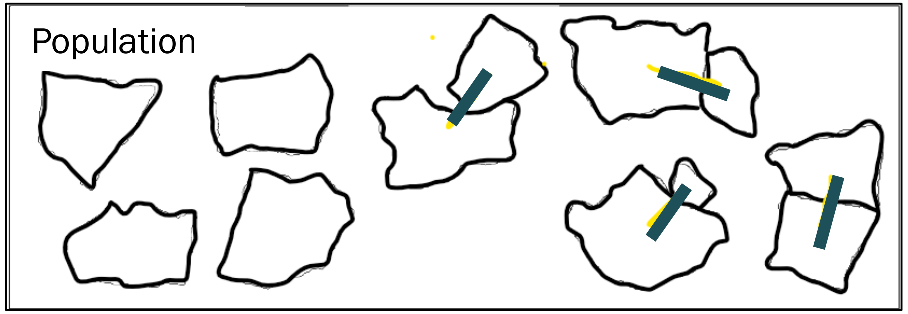
Taken from [6]

# Outline

1. Genetic Algorithms
2. Wall Painting Algorithm
3. Tile Panel Algorithm
4. Conclusions

# Genetic Algorithms (GA)

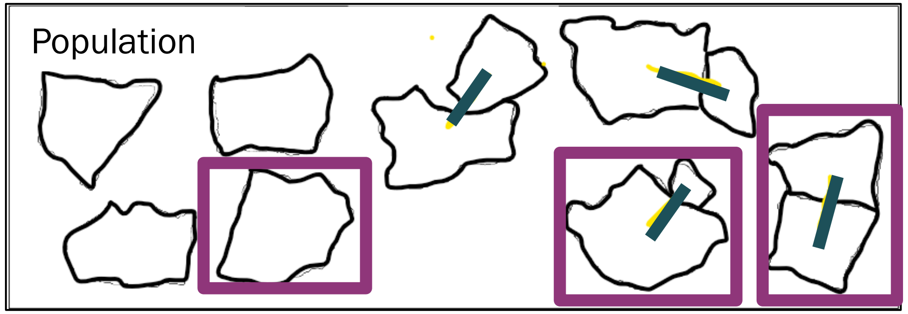
- Based on evolution via natural selection
  - selection, recombination, and mutation
- Selection process: selects solutions made in the previous generation, called parents, to recombine
  - Uses Fitness Functions to determine selection
- Recombination: recombines parents to create new better solutions, called children
- Mutation: adds randomness to the algorithm




Based on figure from [8]

# Genetic Algorithms (GA)

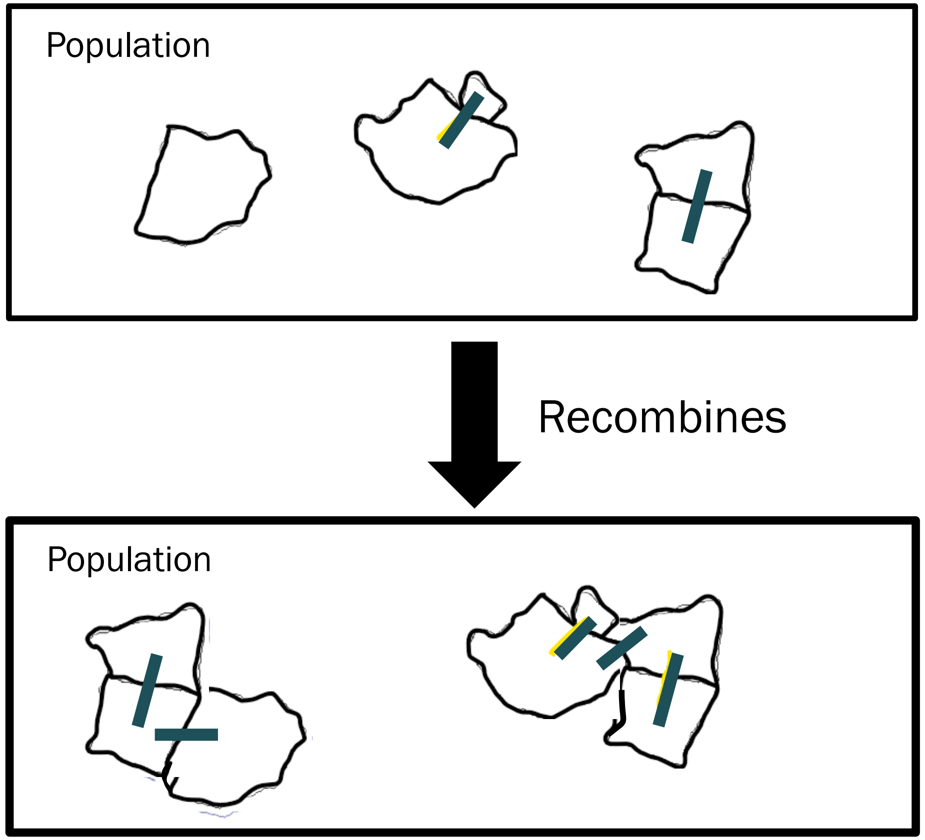
- Based on evolution via natural selection
  - selection, recombination, and mutation
- Selection process: selects solutions made in the previous generation, called parents, to recombine
  - Uses Fitness Functions to determine selection
- Recombination: recombines parents to create new better solutions, called children
- Mutation: adds randomness to the algorithm




Based on figure from [8]

# Genetic Algorithms (GA)

- Based on evolution via natural selection
  - selection, recombination, and mutation
- Selection process: selects solutions made in the previous generation, called parents, to recombine
  - Uses Fitness Functions to determine selection
- Recombination: recombines parents to create new better solutions, called children
- Mutation: adds randomness to the algorithm



# Outline

1. Background
2. **Wall Painting Algorithm**
   - Structure
   - Results
3. Tile Panel Algorithm
4. Conclusions

# Wall Painting Algorithm (WPA)

- Developed by Sizikova and Funkhouser
- GA which given a cluster of fragments, it produces a potential solution
  - A cluster is a group of fragments with matches between them
- Initializes with singleton clusters and paired clusters
  - Singleton clusters: single fragments
  - Paired clusters: two fragments with a match between them

Taken from [8]

# WPA: Selection Procedure

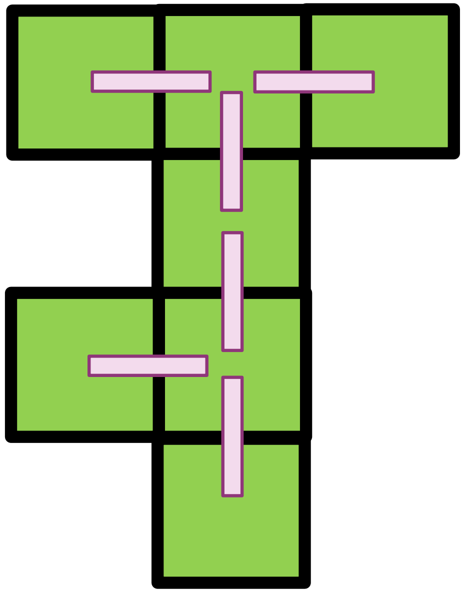
- Starts by ranking clusters with the fitness function developed by Sizikova and Funkhouser
- After the clusters have been ranked, the WPA then filters out clusters with a low number of unique fragments
  - Unique fragments: fragments rarely found in a cluster
  - For example, a fragment found only in one cluster
  - Encourages diversity in clusters

# WPA: Fitness Function

- Goal is to minimize loose connections
- Ranks clusters by calculating the number of fragments, $span_{f_i}$, or the number of matches, $span_{m_i}$, that are part of the spanning tree of cluster, $C_i$
  - $MaxST(C_i)$ is the sum of the match scores of the maximal spanning tree of cluster, $C_i$
  - W is a weighting parameter

$$f(c_i) = MaxST(C_i) - W(span_{f_i} + span_{m_i})$$

Spanning Tree

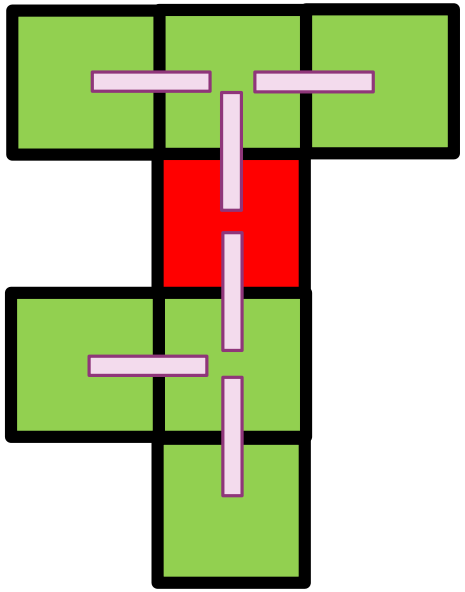


# WPA: Fitness Function

- Goal is to minimize loose connections
- Ranks clusters by calculating the number of fragments, $span_{f_i}$, or the number of matches, $span_{m_i}$, that are part of the spanning tree of cluster, $C_i$
  - $MaxST(C_i)$ is the sum of the match scores of the maximal spanning tree of cluster, $C_i$
  - W is a weighting parameter

$$f(c_i) = MaxST(C_i) - W(span_{f_i} + span_{m_i})$$

Spanning Tree

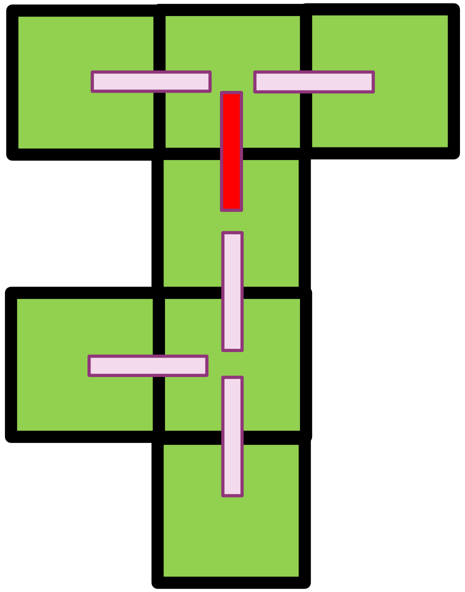


# WPA: Fitness Function

- Goal is to minimize loose connections
- Ranks clusters by calculating the number of fragments, $span_{f_i}$, or the number of matches, $span_{m_i}$, that are part of the spanning tree of cluster, $C_i$
  - $MaxST(C_i)$ is the sum of the match scores of the maximal spanning tree of cluster, $C_i$
  - W is a weighting parameter

$$f(c_i) = MaxST(C_i) - W(span_{f_i} + span_{m_i})$$

Spanning Tree

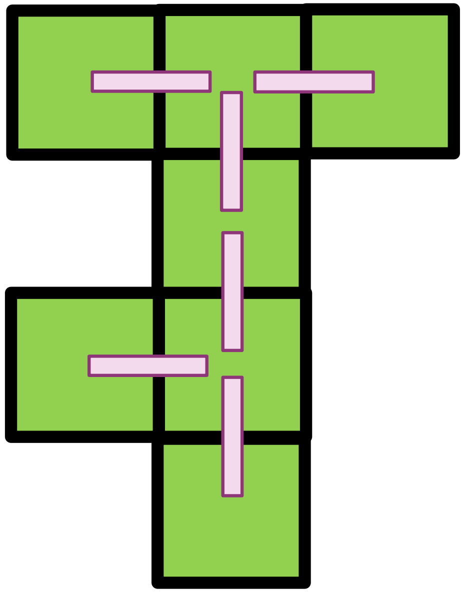


# WPA: Fitness Function

- Goal is to minimize loose connections
- Ranks clusters by calculating the number of fragments, $span_{f_i}$, or the number of matches, $span_{m_i}$, that are part of the spanning tree of cluster, $C_i$
  - $MaxST(C_i)$ is the sum of the match scores of the maximal spanning tree of cluster, $C_i$
  - W is a weighting parameter

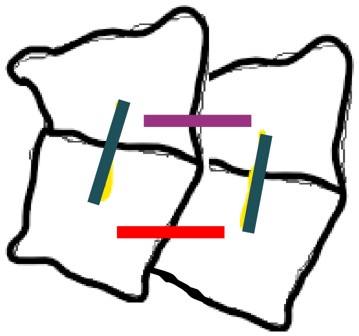
$$f(c_i) = MaxST(C_i) - W(span_{f_i} + span_{m_i})$$

Spanning Tree



# WPA: Match Scores

- When a match is being considered, the WPA will score it as $C_{ik}(1 + 0.1M)$
  - $C_{ik}$ is the fitness score of the resulting cluster
  - $M$ is the number of matches being added



# WPA: Recombination Process

- Recombines through two methods: by fragment or by match
- When combining by fragment, both parents must share a fragment
  - WPA considers all possible spanning fragments and ways they connect, choosing the cluster with the highest match score
- When combining by match, produces a match between the parents
  - Uses a weighted probability to choose which matches will be considered

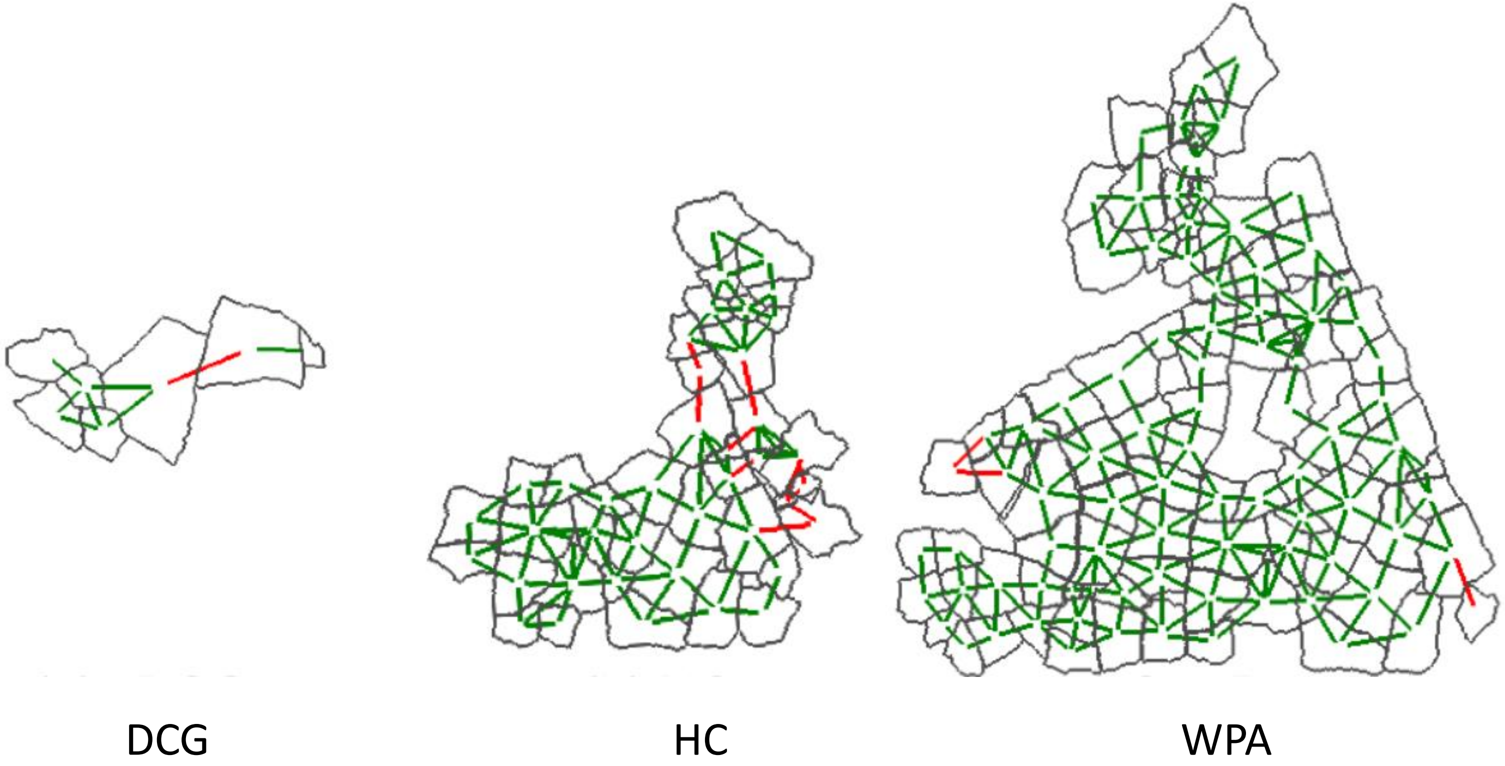
$$P(i) = \frac{f_i}{\sum_{k=1}^{N} f_k}$$

# WPA: Results

- A fresco was created and artificially fragmented and weathered for testing algorithms in this problem space
- Compared WPA  to three other algorithms: dense cluster growth (DCG), hierarchical clustering (HC), and the previous state of the art algorithm developed by Castañeda et al.

# WPA: Results

- When comparing the DCG and HC to WPA, Sizikova and Funkhouser used the same initial data
- F-Score is the average of precision and recall
  - Precision: proportion of correct matches in a solution
  - Recall: proportion of correct matches out of the whole painting




Taken from [8]

| Method | # of Fragments | F-Score |
|---|---|---|
| WPA | 90 | 0.823 |
| HC | 42 | 0.411 |
| DCG | 7 | 0.082 |

Taken from [8]

# WPA: Results

- Sizikova and Funkhouser compared the WPA visually against Castañeda et al.

Castañeda et al.

WPA

Taken from [8]

# Outline

1. Background
2. Wall Painting Algorithm
3. **Tile Panel Algorithm**
    - Deep Learning
    - GA Structure
    - Results
4. Conclusions

# Tile Panel Algorithm (TPA)

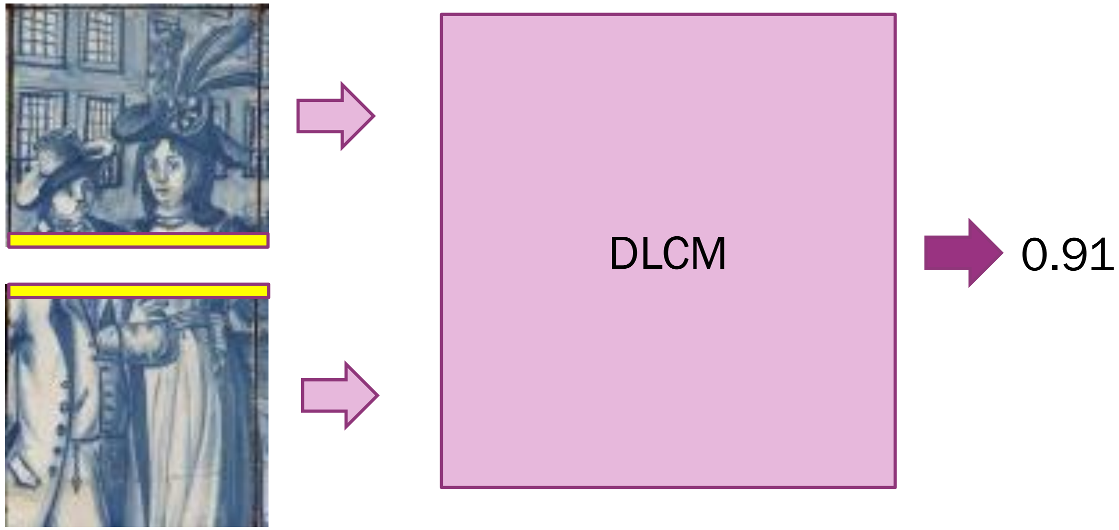
- Developed by Rika et al.
- Combination of a Kernel Growth GA and a Deep Learning Compatibility Measure
- Compatibility Measure: determines if two given edges match
- Used human expert reconstructed tile panels for testing

# Deep Learning (DL)

- Type of Machine Learning based on Neural Networks
- Uses 4 networks to analyze the color channels (RGB)
  - One for each separately and one together
  - Each returns a compatibility score, which is then added for the overall score

# Deep Learning Compatibility Measure (DLCM)

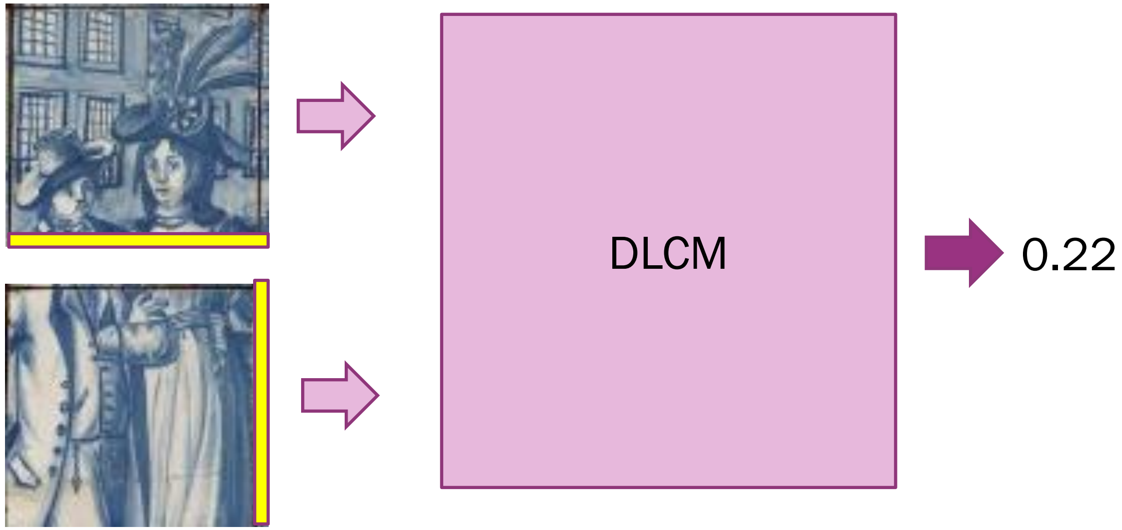
- Given two edges, returns a real number to signify its compatibility as an adjacent piece




Tiles from https://artsandculture.google.com/asset/the-hat-maker-panels-detail-real-f%C3%A1brica-de-lou%C3%A7a-ao-rato/kgG8platOk9fWA

# Deep Learning Compatibility Measure (DLCM)

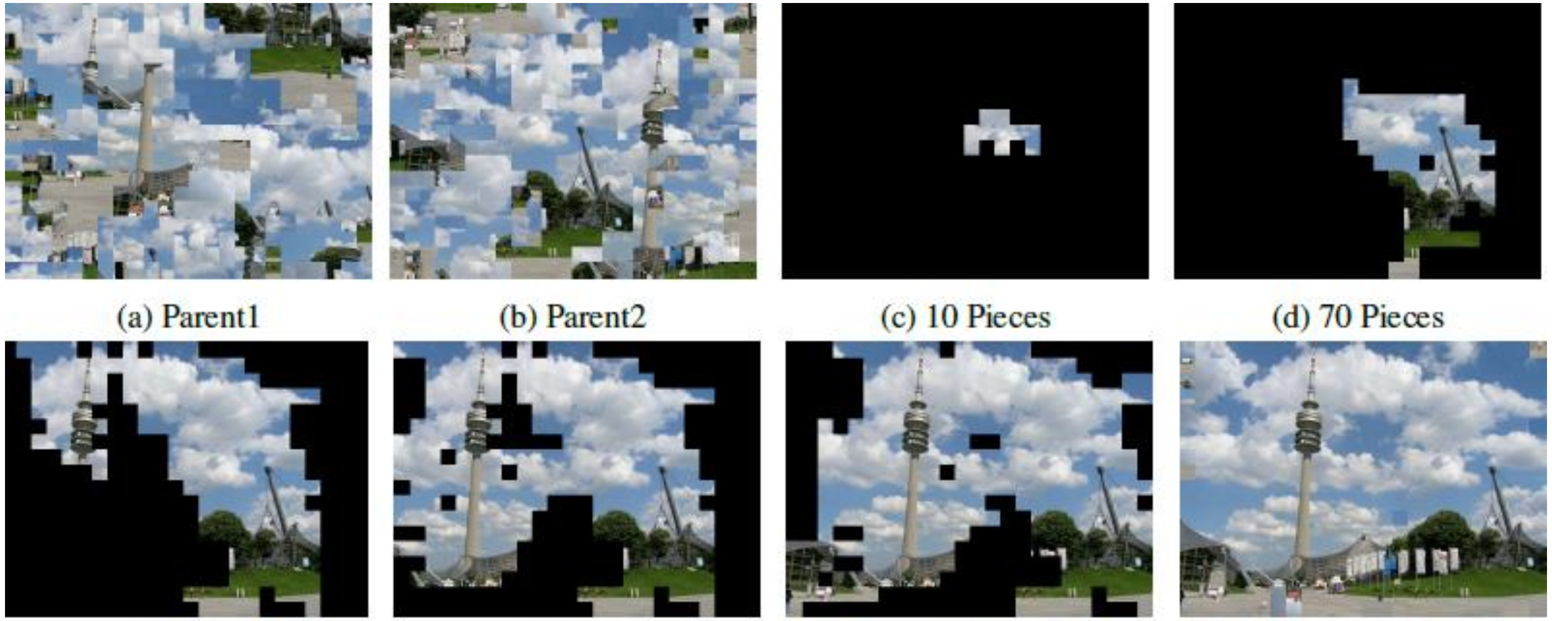
- Given two edges, returns a real number to signify its compatibility as an adjacent piece




Tiles from https://artsandculture.google.com/asset/the-hat-maker-panels-detail-real-f%C3%A1brica-de-lou%C3%A7a-ao-rato/kgG8platOk9fWA

# Original Kernel-Growth GA

- Kernel: a portion of assembled pieces
- Takes a kernel and adds pieces to the adjacent edges of it

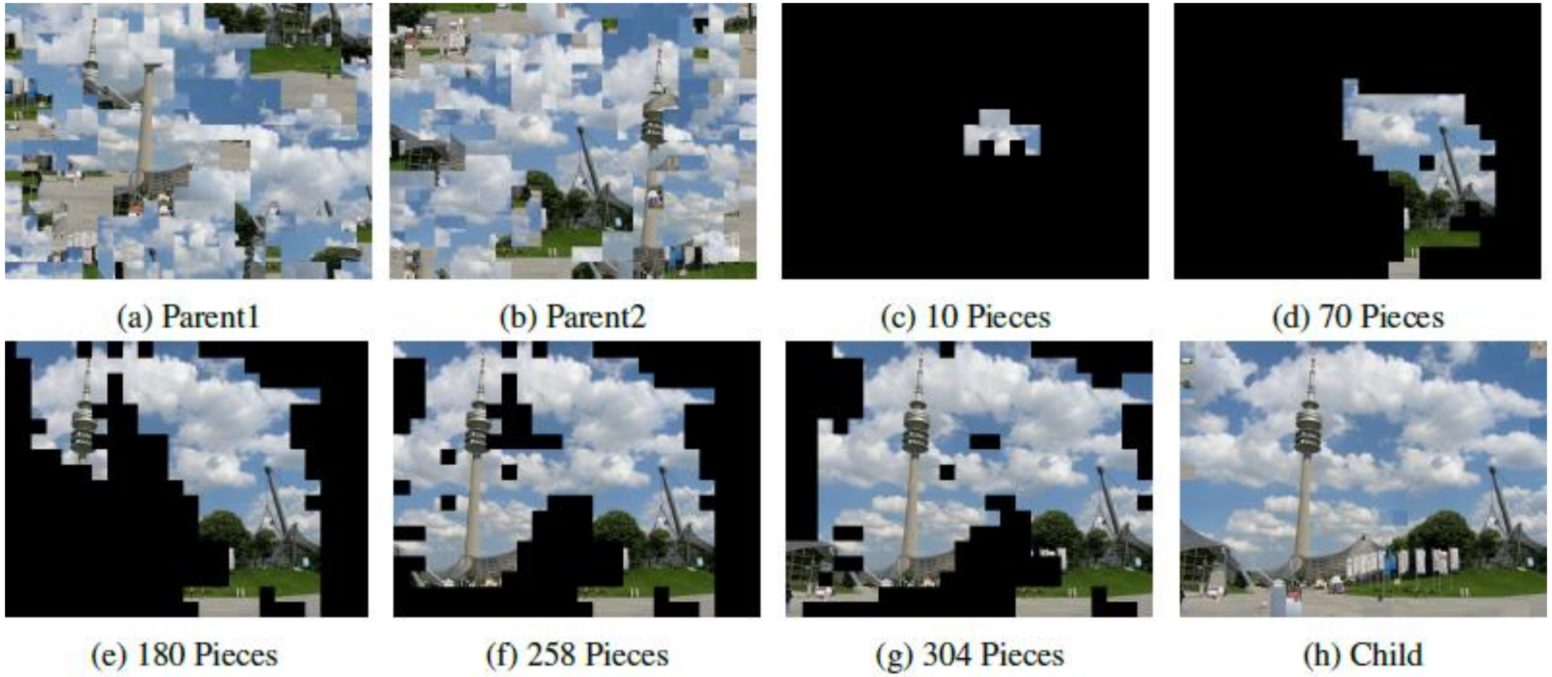


(a) Parent1 (b) Parent2 (c) 10 Pieces (d) 70 Pieces
(e) 180 Pieces (f) 258 Pieces (g) 304 Pieces (h) Child

Taken from [7]

# TPA: Kernel-Growth GA

- Uses 6 hierarchical phases to decide which piece gets added
- If one phase fails, moves on to the next until a success, and repeats process for each new addition



(a) Parent1 (b) Parent2 (c) 10 Pieces (d) 70 Pieces

(e) 180 Pieces (f) 258 Pieces (g) 304 Pieces (h) Child

Taken from [7]

# TPA: Hierarchical Phases

- Phase I: Adds piece from parent with higher finess with average compatibility measure between it and all of its neighbors greater than *max(0.8, $C_{mean}$)*
  - $C_{mean}$ is the parent's average compatibility score across all tile edges
- Phase II: Similar to Phase I, but it selects a piece from the parent with the lower fitness score

(a) Parent1

(b) Parent2

Based on Figure from [7]

Before

After

# TPA: Hierarchical Phases

- Phase III: Adds piece that both parents agree is adjacent to the edge

(a) Parent1 (b) Parent2

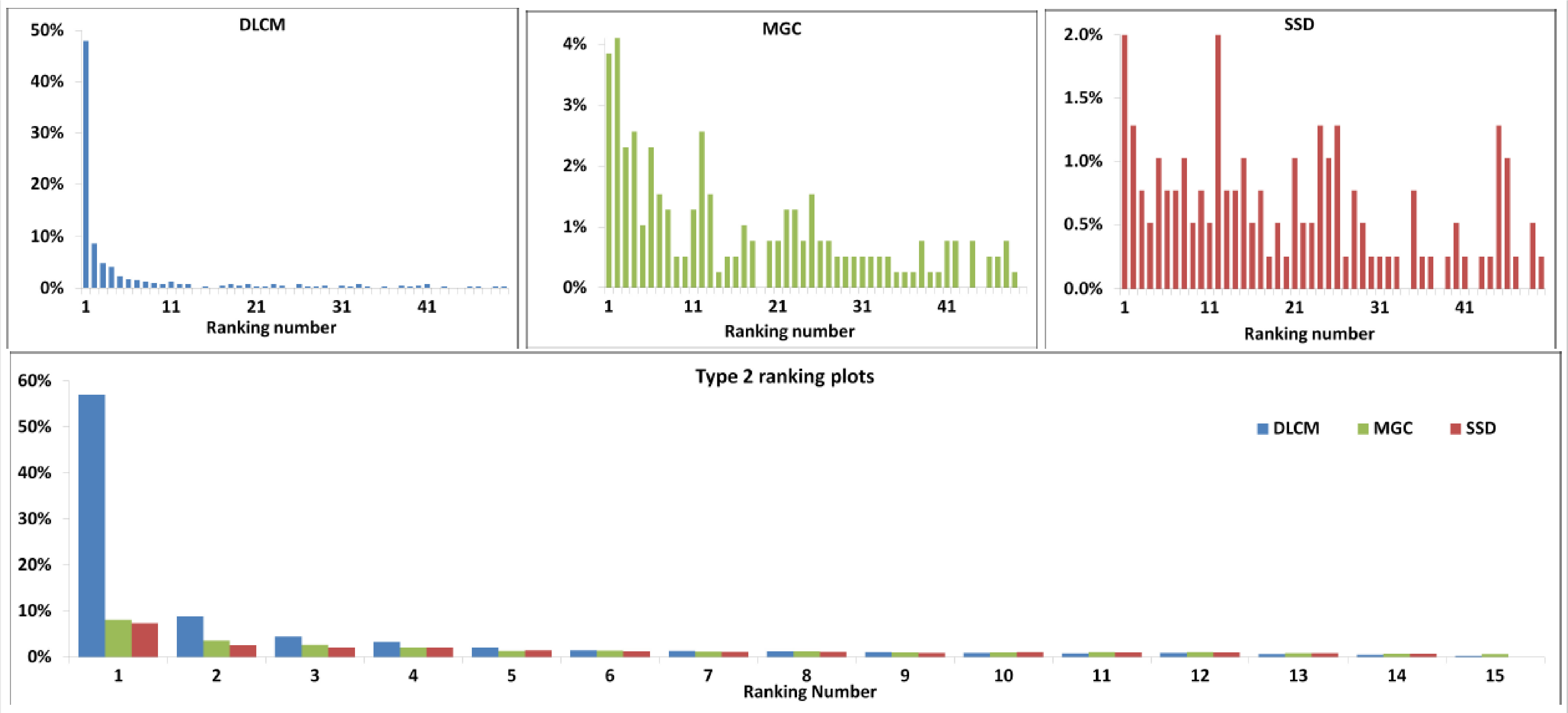
Based on Figure from [7]

# TPA: Hierarchical Phases

- Phase IV: adds the most compatible piece, if available
- Phase V: adds the second most compatible piece, if available
- Phase VI: adds a random piece

# TPA: Results

- Compared just the DLCM against sum of squared differences (SSD) and Mahalanobis gradient compatibility (MGC)
- Rika et al. compared whole TPA against
  - Gallagher's algorithm combined with MGC
  - Original Kernel Growth algorithm combined with their proposed DLCM
- Used two different test cases: Type 1, with known orientation, and Type 2, with unknown orientation


DLCM
50%
40%
30%
20%
10%
0%
1
11
21
31
41
Ranking number
MGC
4%
3%
2%
1%
0%
1
11
21
31
41
Ranking number
SSD
2.0%
1.5%
1.0%
0.5%
0.0%
1
11
21
31
41
Ranking number
Type 2 ranking plots
60%
50%
40%
30%
20%
10%
0%
DLCM
MGC
SSD
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Ranking Number

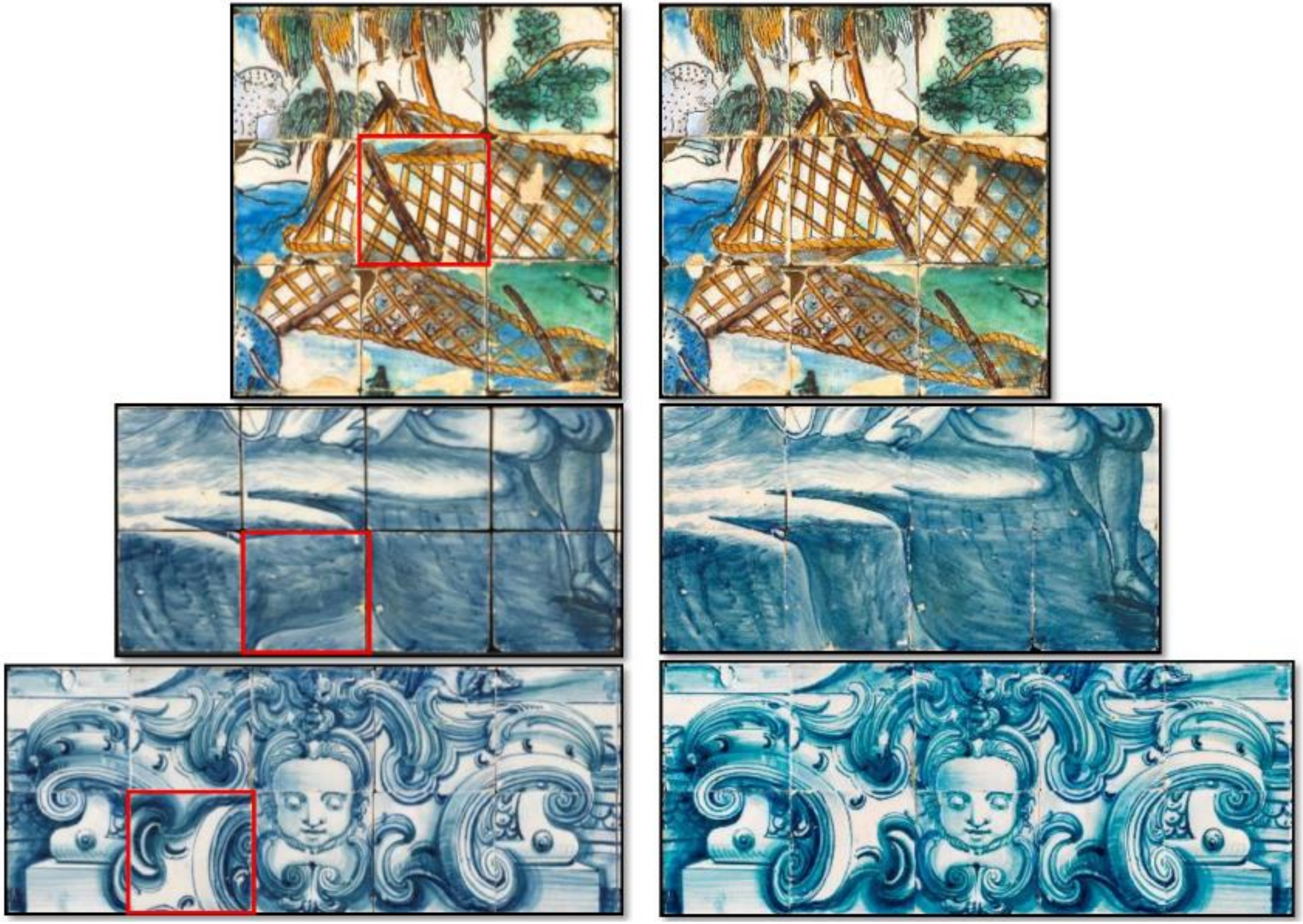
Taken from [6]

# TPA: Results

| Method | Type 1 | | Type 2 | |
|---|---|---|---|---|
| | Known Dims. | Unknown Dims. | Known Dims. | Unknown Dims. |
| Gallagher + MGC | – | 13.0% | – | 3.5% |
| Kernel-growth + DLCM | 84.5% | – | 58.6% | – |
| TPA (using DLCM) | 96.3% | 96.0% | 86.8% | 82.2% |

% of correct matches

Based on Table from [6]



Taken from [6]

# Conclusions

- More can still be done to improve algorithms in this problem space
- Wall Painting Algorithm (WPA)
  - Apply to unsolved wall paintings
- Tile Panel Algorithm (TPA)
  - Account for missing tiles
  - Be able to deal with pool of tiles from more than one panel

# Questions?

# References


[1] Convolutional neural network, Oct 2019. https://en.wikipedia.org/wiki/Convolutional_neural_network.

[2] Deep learning, Oct 2019. https://en.wikipedia.org/wiki/Deep_learning.

[3] Genetic algorithm, Oct 2019. https://en.wikipedia.org/wiki/Genetic_algorithm.

[4] A. G. Casañeda, B. J. Brown, S. Rusinkiewicz, T. A. Funkhouser, and T. Weyrich. Global consistency in the automatic assembly of fragmented artefacts. In VAST, volume 11, pages 73–80, 2011.

[5] A. C. Gallagher. Jigsaw puzzles with pieces of unknown orientation. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 382–389, Los Alamitos, CA, USA, jun 2012. IEEE Computer Society.

[6] D. Rika, D. Sholomon, E. O. David, and N. S. Netanyahu. A novel hybrid scheme using genetic algorithms and deep learning for the reconstruction of portuguese tile panels. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19, pages 1319–1327, New York, NY, USA, 2019. ACM.

[7] D. Sholomon, O. David, and N. S. Netanyahu. A genetic algorithm-based solver for very large jigsaw puzzles. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013.

[8] E. Sizikova and T. Funkhouser. Wall painting reconstruction using a genetic algorithm. J. Comput. Cult. Herit., 11(1):3:1–3:17, Dec. 2017