

# Solving multiple square jigsaw puzzles with missing pieces

Dominique Cheray and Manuel Krämer

March 27, 2018

## Abstract

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Materials and Methods</b>	<b>3</b>
2.1	Algorithm . . . . .	3
2.1.1	Compatibility between pieces . . . . .	3
2.1.2	First piece . . . . .	4
2.1.3	Placer . . . . .	4
2.2	Image data . . . . .	4
<b>3</b>	<b>Results</b>	<b>5</b>
<b>4</b>	<b>Discussion</b>	<b>6</b>

# Chapter 1

## Introduction

## Chapter 2

# Materials and Methods

### 2.1 Algorithm

The algorithm we implemented is the one that was used in [1]. The whole procedure can be described by three main steps:

1. Compatibility between pieces: This is done by calculating the dissimilarity between two pieces and the second best dissimilarity. With this score one can determine the so called "Best Buddies"-metric which indicates, that two pieces agree that each other is their best neighbor.
2. First piece: A good piece to start with has best buddies in all four spatial relations and these four neighbors have best buddies in all directions as well.
3. Placer: The placing of the pieces is done by grabbing the best piece (highest mutual compatibility) from a pool, placing this piece and adding the best buddies of it to the pool.

#### 2.1.1 Compatibility between pieces

The most fundamental part is the calculation of the compatibility between two pieces to determine how well they fit together. This is done by computing the dissimilarity between every pair of pieces and the confidence in this score.

First of all, the dissimilarity of two pieces  $p_i$  and  $p_j$  is defined as follows (see [1]):

$$D(p_i, p_j, right) = \sum_{k=1}^K \sum_{d=1}^3 ||((2p_i(k, K, d) - p_i(k, K - 1, d)) - p_j(k, 1, d))|| \quad (2.1)$$

$K$  is the piece size and  $d$  the dimension.

By just considering a small dissimilarity as a measure of adjacency the algorithm would face the problem, that this value is very high in distinctive regions.

To deal with this fact we also calculate the dissimilarity of the second best neighbor; if these two values differ, the closest pieces are more likely to be neighbors. With that we can define the compatibility as proposed in [1]:

$$C(p_i, p_j, r) = 1 - \frac{D(p_i, p_j, r)}{\text{second}D(p_i, r)} \quad (2.2)$$

This equation expresses how reliable a match of two pieces is. If now two pieces consider each other as their most likely neighbor in a certain spatial relation, these two are called best buddies.

### 2.1.2 First piece

One important task is to avoid early errors because the algorithm considers the unplaced and placed pieces as well. If there is a mistake within the first few pieces this can lead to following errors. This means, we need a first piece that is located in a distinctive area where the probability of making errors is as low as possible. A good measure for this is the best buddies metric. We require the first piece to have best buddies in all four spatial relations, what ensures that the piece itself is distinctive. Furthermore, all four neighbors of this piece need to have best buddies in all four spatial relations as well, what means that the piece is located in a distinctive region. Usually there are more than one piece that satisfy these conditions. Out of all these distinctive pieces we choose the one with the best buddies that maximize the mutual compatibility in all four spatial relations (as defined in [1]):

$$\tilde{C}(p_i, p_j, r_1) = \frac{C(p_i, p_j, r_1) + C(p_j, p_i, r_2)}{2} \quad (2.3)$$

### 2.1.3 Placer

## 2.2 Image data

## Chapter 3

# Results

## Chapter 4

# Discussion



# Bibliography

- [1] G. Paikin and A. Tal, “Solving multiple square jigsaw puzzles with missing pieces,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 4832–4839, 2015.
- [2] T. S. Cho, S. Avidan, and W. T. Freeman, “A probabilistic image jigsaw puzzle solver,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 183–190, 2010.
- [3] D. Sholomon, O. David, and N. S. Netanyahu, “A Generalized Genetic Algorithm-Based Solver for Very Large Jigsaw Puzzles of Complex Types,” *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, vol. 28, no. 4, pp. 2839–2845, 2014.
- [4] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, “A fully automated greedy square jigsaw puzzle solver,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 9–16, 2011.
- [5] X. Yang, N. Adluru, and L. J. Latecki, “Particle filter with state permutations for solving image jigsaw puzzles,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2873–2880, IEEE, 2011.
- [6] A. C. Gallagher, “Jigsaw puzzles with pieces of unknown orientation,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 382–389, IEEE, 2012.
- [7] D. Sholomon, O. David, and N. S. Netanyahu, “A genetic algorithm-based solver for very large jigsaw puzzles,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 1767–1774, IEEE, 2013.
- [8] K. Son, J. Hays, and D. B. Cooper, “Solving square jigsaw puzzles with loop constraints,” in *European Conference on Computer Vision*, pp. 32–46, Springer, 2014.