Implementing Camera Pose Estimation by Grey Wolf Optimizer

Artificial Intelligence

Ferdowsi University of Mashhad

[transcendentimagination@gmail.com](mailto:transcendentimagination@gmail.com)

Abstract

In this report, we’re going to improve the camera pose estimation from two images taken by two uncalibrated cameras by the Grey Wolf Optimizer algorithm. First we estimate random camera poses by the five point algorithm as the initial population and then we try to find the optimum solution with the Grey Wolf Optimizer. This is an effort to implement the required algorithm and fitness calculations according to the Elashry and Toth [1] that introduces this method.

Keywords

Camera Pose Estimation, Essential Matrix, Fundamental Matrix, Five Point Algorithm, Grey Wolf Optimizer, Swarm Particle Algorithm

Table of contents

[Implementing Camera Pose Estimation by Grey Wolf Optimizer 1](#_Toc171703079)

[Abstract 1](#_Toc171703080)

[Keywords 1](#_Toc171703081)

[Table of contents 2](#_Toc171703082)

[Table of figures 2](#_Toc171703083)

[1- Introduction 3](#_Toc171703084)

[2- Improving Camera Pose Estimation Using Swarm Particle Algorithms 3](#_Toc171703085)

[3- An Efficient Solution to the Five-Point Relative Pose Problem 3](#_Toc171703086)

[4- Grey Wolf Optimizer 3](#_Toc171703087)

[5- Fitness Function 3](#_Toc171703088)

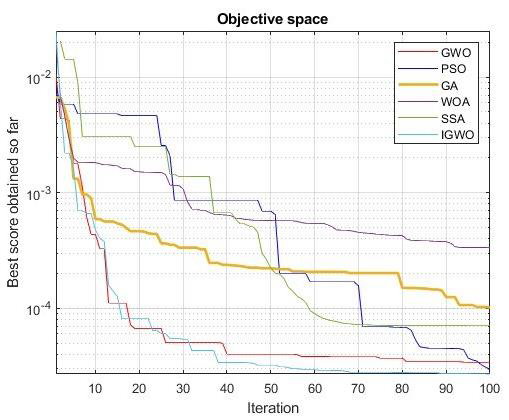
[6- Implementation 4](#_Toc171703089)

[7- Results 4](#_Toc171703090)

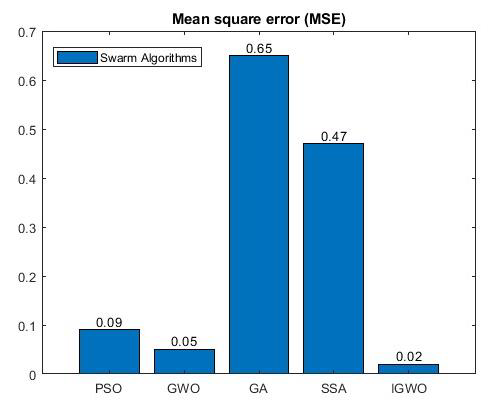
[8- Conclusion 4](#_Toc171703091)

[9- References 4](#_Toc171703092)

Table of figures

[ 7](#_Toc171704062)

[Figure 1: Results obtained from metaheuristic optimization algorithms. 7](#_Toc171704063)

[ 8](#_Toc171704064)

[Figure 2: Histogram of comparing between the algorithms 8](#_Toc171704065)

[A graph with numbers and a number of blue bars

Description automatically generated with medium confidence 9](#_Toc171704066)

[Figure 3: Histogram of computational time for the different algorithms 9](#_Toc171704067)

[Figure 1: Camera Pose Fitness Function 10](#_Toc171704068)

# Introduction

First, we introduce the articles used for this work and then we explain the implementations for the algorithms. Finally, we test and compare the results. The problems is follows: We have 2 images captured by the same or different cameras at different positions and rotations (poses). We want to estimate the relative orientation between these two poses from the 2 captured images. In this problem, we know the intrinsic parameters of the cameras, but we don’t have any knowledge about the scene. The general flow is like this:

* Estimate some camera poses with five-point algorithm as the initial solutions of the swarm algorithm. We do it by repeating the five-point algorithm multiple times by choosing random matched points.
* Then, we apply the Grey Wolf Optimizer over the solutions until certain iterations. We choose the alpha wolf as the best solution.

For finding the feature points and matching them, we use the SURF algorithm. For the fitness function we use an innovative way the calculated the distances of the match points on the assumed second camera

# Improving Camera Pose Estimation Using Swarm Particle Algorithms

Camera pose estimation has been a common problem in the field of computer vision. There have been introduced many methods to solve it. They all propose valid solutions and estimate good poses for the cameras, but they may or may not be good enough. The proposed method in that article aims to improve those solutions by evolutionary algorithms (EA) or swarm algorithms (SA) to find the optimum solution in the search space.

It first describes some of the EA and SA algorithms that it wants to apply on the pose estimation including the Genetic Algorithm, Particle Swarm Optimization, Grey Wolf Optimization, Improved Grey Wolf Optimization, and Whal Optimization. Then, it compares the results of these algorithms on the SPIN Lab dataset by best obtained score, mean square error, and computation time. Finally, it concludes that the IGWO performs best in finding the optimum solution and requires most computation time.

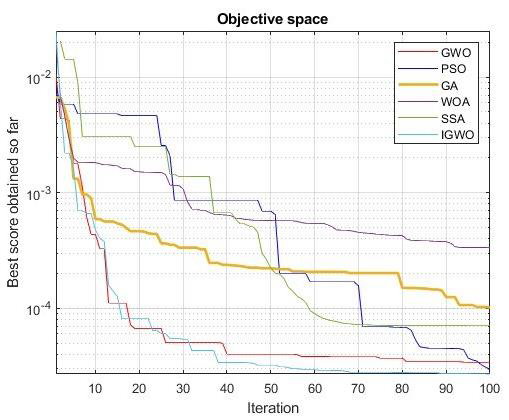


Figure 1: Results obtained from metaheuristic optimization algorithms.

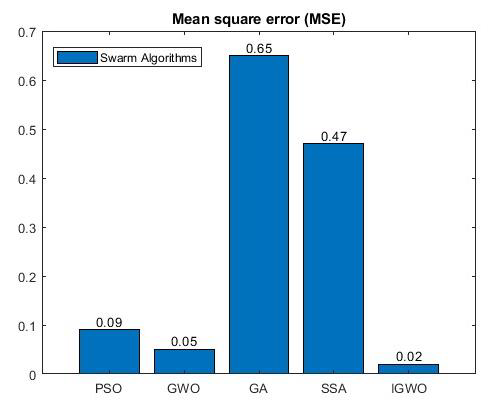


Figure 2: Histogram of comparing between the algorithms

A graph with numbers and a number of blue bars

Description automatically generated with medium confidence

Figure 3: Histogram of computational time for the different algorithms

# An Efficient Solution to the Five-Point Relative Pose Problem

This article tries to find the essential matrix by giving five matched points between two images and knowing the intrinsic parameters of the cameras. It solves a tenth degree polynomial equation to find the coefficients.

Note that in this report, we use a different five-point algorithm from what the Elashry and Toth [1] used in its experiments.

# Grey Wolf Optimizer

This algorithm aims to simulate the hunting behavior of the grey wolfs who move in a pack usually consists of 5-12 members. It considers each wolf as candidate solution and the prey as the best solution trying to find. It first categorizes the wolfs into a hierarchy of ranks that determines how the wolfs will move in the search space. There are leader wolfs from the top to the bottom named alpha, beta, and delta wolfs. There is only one of each rank. They are considered as the best three solutions found so far. These leaders are assumed to know the most probable position for the prey. They are responsible for pursuing the solution and leading the pack. The other wolfs are called omega and the move according to the movement of the leaders and some randomness. They follow this pattern until they encircle and hunt the prey.

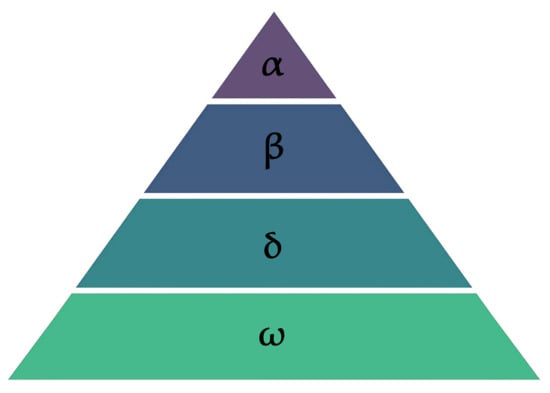


Figure 4: Hierarchy of grey wolf (dominance decreases from top down)

The method consists of different aspects:

## Encircling

The wolves try to close the prey. So, they need to decrease their distance from the prey over time. It controls this closing with A and C vectors. The Xp is the position of the prey and the X(t) is the position of the wolf. On each iteration, the wolf updates its position to get closer to the prey.

## Hunting

## Attacking

## Searching

## Algorithm

# Fitness Function

To give a fitness value to each candidate relative pose, we need to define a value that describes how accurate is this pose if it was the pose for the second camera. The matched points found by the SURF algorithm tell us that they are the same points in the 3D space. So, we simulate reconstructing those points in the 3D space and examine how feasible it is. For each matched point in the first camera, we find a ray emitting from the camera origin toward a that pixel on hypothetical plane in front of that camera in the pinhole camera model. This ray will cross the real point the 3D space. We do the same for that matched point on the second image with assumed second camera pose. Then, we determine how close these to rays are by finding the distance between those lines. We repeat this process for all matched points and give the sum of those distances as the fitness value for that candidate relative pose (solution). Because the GWO algorithm tries to find a solution with the minimum fitness, the distance sum must also be minimized.

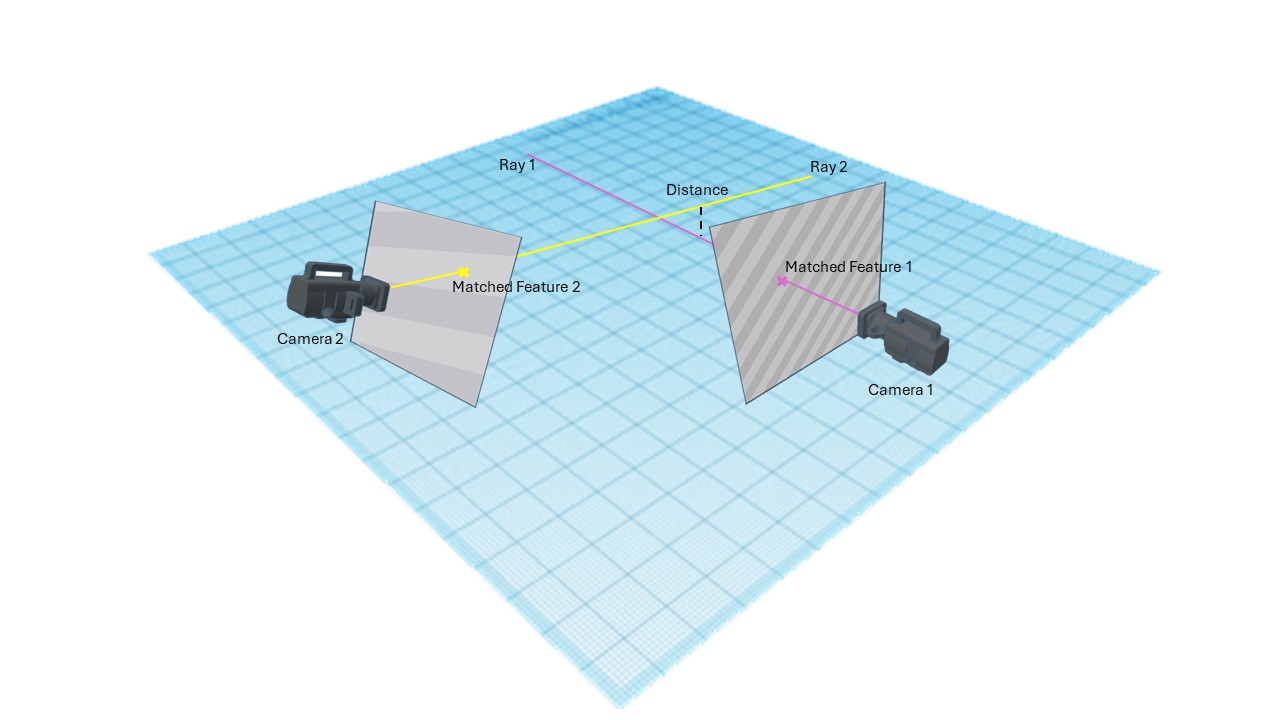


Figure 1: Camera Pose Fitness Function

# Implementation

After

# Results

# Conclusion

We saw that the camera pose estimation for 2 uncalibrated cameras with 2 images can be further improved by applying the Grey Wolf Optimizer on the random initial solutions and finding the optimum solution.

# References

Elashry, A. and C. Toth (2023). "IMPROVING CAMERA POSE ESTIMATION USING SWARM PARTICLE ALGORITHMS." Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. **XLVIII-M-3-2023**: 87-93.