An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems

A 30% Project report submitted to Rajagiri School of Engineering & Technology in partial fulfilment of degree of

B.Tech.

in

Electronics & Communication Engineering

by

Varsha Ramdas

Roll No: RET17EC177

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

KERALA

JANUARY 2021

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

KOCHI - 682039,

Certificate

*This is to certify that this report entitled "An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems" is a bonafide record of the 30% project report presented by **Ms. Varsha Ramdas, Roll No. RET17EC177** under our guidance towards the partial fulfilment of the requirements for the award of **Bachelor of Technology in Electronics & Communication Engineering** of the **APJ Abdul Kalam Technological University**.*

Mr. Abhishek Viswakumar	**Dr. Rithu James**
Assistant Professor	Head of Department
Dept. of ECE	Dept. of ECE
RSET	RSET
Kochi	Kochi

Acknowledgement

I am extremely grateful to **Prof.(Dr.) Sreejith P S**, Principal, Rajagiri School of Engineering and Technology, and **Dr. Rithu James**, Head of the Department, Electronics and Communication Engineering, for providing all the required resources for the successful completion of my 30% project. My heartfelt gratitude to my project guide **Mr. Abhishek Viswakumar, Asst. Professor** and project coordinators, **Mr. Naveen N, Ms. Mariya Vincent and Dr. Simi Zerine Sleeba** for their valuable suggestions and guidance. I express my sincere thanks to all staff members and friends for their help and co-ordination in bringing out this 30% project successfully in time. I also am grateful to the authors of the references and other literatures referred to in this project. Thank you.

Abstract

Grey wolf optimizer (GWO) is a new meta-heuristic algorithm. It mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. An improved grey wolf optimizer based on tracking and seeking mode. Proposed to improve the diversity of the population and the ability of the algorithm to balance exploration and exploitation. Four types of grey wolves such as alpha, beta, delta, and omega are employed for simulating the leadership hierarchy. In addition, the three main steps of hunting, searching for prey, encircling prey, and attacking prey, are implemented. Grey Wolves are apex predators and they perform division of labour to get hold of predators. A mathematical modelling, simulation and a comparative study performed between various meta heuristic algorithms including Particle swarm Optimization (PSO), Sine Cosine Algorithm (SCA), Ant Lion Optimizer (ALO) and Moth Flame Optimization (MFO). Simulation and analytical results based on 30 benchmark functions including uni modal and multimodal functions show the superior exploitation, exploration and convergence of the algorithm to optimal value zero.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The project is based on Swarm Robotics. A robot swarm is a self-organizing multi-robot system characterized by high redundancy. Robots sensing and communication capabilities are local and robots do not have access to global information. The collective behaviour of the robot swarm emerges from the interactions of each individual robot with its peers and with the environment. Typically, a robot swarm is composed of homogeneous robots, although some examples of heterogeneous robot swarms do exist.

The objective of the project is to conduct a comparative study of simulation of bots under novel nature inspired algorithms and their performance and the implementation of a centralized shape forming system.

The project involves self organization of robots using Computer Vision and Algorithm Analysis. Grey Wolf Algorithm, Particle Swarm Optimization algorithm, ... were analysed on the basis of convergence. Elapsed time was also analysed as a benchmark for various unimodal and multimodal functions. The object tracking and path planning is done using OpenCV with the help of Aruco markers.

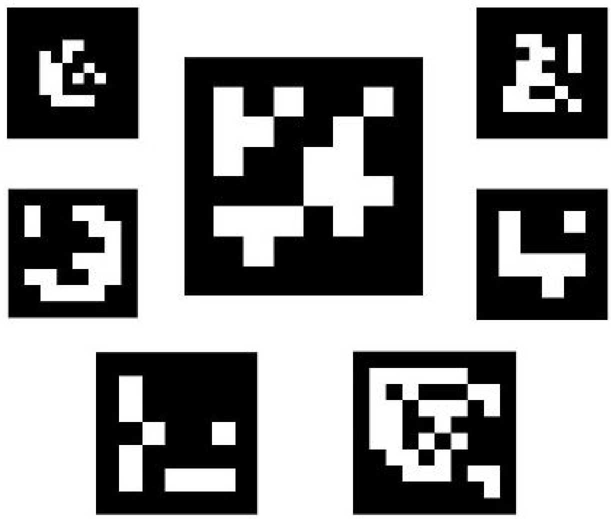

Figure 1.1: Aruco Marker images

An ArUco marker is a synthetic square marker composed by a wide black border and an inner binary matrix which determines its identifier (id). The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques. This helps estimate pose and position of the robot with respect to frame.

A NodeMCU microcontroller with inbuilt Wi-Fi module is used to enable communication between the master and the slaves.

Chapter 2

Work Done

As part of the 30 percent completion of the project, the literature review part and subsequently the seminars were conducted. This involved the thorough study of a research paper on various topics related to the project. Each member of the group presented a seminar based on such research papers studied as part of the literature review.

The work on the hardware as well as algorithm analysis/simulation parts of the project were also started.

2.1 Progress on Hardware Prototype

Work on implementing the hardware part of the project was started. Figure 2.1 shows the block diagram of the prototype.

The following are the topics completed as part of 30% project completion:

- Camera calibration.

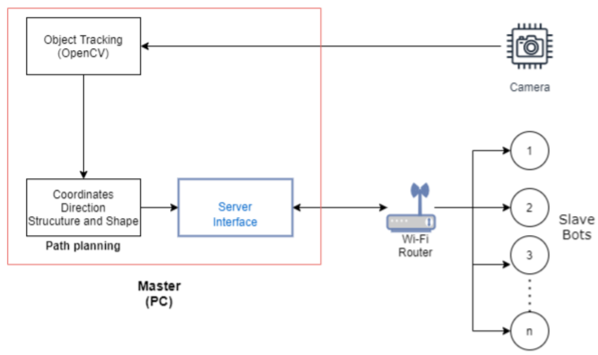

Figure 2.1: Block Diagram

- Obtaining position and orientation from ArUco marker.

2.1.1 Camera Calibration

Camera calibration consists in obtaining the camera intrinsic parameters and distortion coefficients. This parameters remain fixed unless the camera optic is modified, thus camera calibration only need to be done once.

The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system. The intrinsic parameters represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates. This is shown in figure 2.2

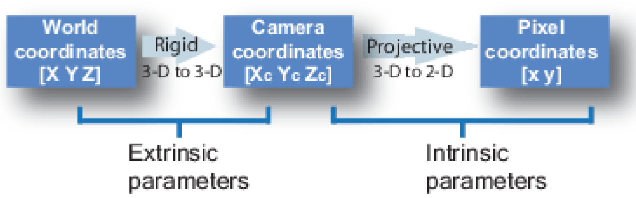

Figure 2.2: Camera Calibration

Camera calibration is usually performed using the OpenCV `calibrateCamera()` function. This function requires some correspondences between environment points and their projection in the camera image from different viewpoints. In general, these correspondences are obtained from the corners of chessboard patterns.

Using the ArUco module, calibration can be performed based on ArUco markers corners or ChArUco corners. Calibrating using ArUco is much more versatile than using traditional chessboard patterns, since it allows occlusions or partial views.

Camera calibration for the camera used for the project was done using chessboard pictures. Around 50 images of the chessboard at various angles and distances were captured using the camera and the images were used to create the camera calibration file. The camera calibration was hence succesfully done and the calibrations values were stored for future use in the project.

Chapter 3

Conclusion

[?] [?] [?] [?] [?]