

A Parallel Real Time multiple Objects Tracking Using Particle Swarm Optimization Algorithm

Ahlam Hasan Albashiti
Supervisor: Mohammed Al-Dasht
Master of Informatics
College of Graduate Studies
Palestine Polytechnic University

Introduction

Multiple objects tracking is to track multiple moving objects in video sequences.

It is a challengeable field of the computer vision science; because of the illumination changing which affects the color intensity, the complexity of the object shape and motion, and the object occlusion.

Particle swarm Optimization (PSO) algorithm is suitable for real time cases, object transformation, and work under dynamic environment, and illumination changes.

In this project we build a master-slave system, we depend on domain decomposition, where the master takes the objects and distributes them to slaves, then each slave tracks its object using PSO algorithm, and returns the new position of the object. Color histogram is used to model the object to be tracked.

Project Objectives:

The main contribution of this paper is to track multiple general purpose objects defined by the user regardless of their shape, color, size, and direction of motion in parallel processors.

This work is the first for tracking parallel multiple objects using PSO algorithm. We are the first by the idea each processor tracks its own objects.

Design and Implementation

1.Decomposition : the data is decomposed and is distributed among slaves. The master distributes the user defined objects to slaves. After that, each slave tracks its object, and then returns the coordinates for all particles to the master to be drawn on the frame.

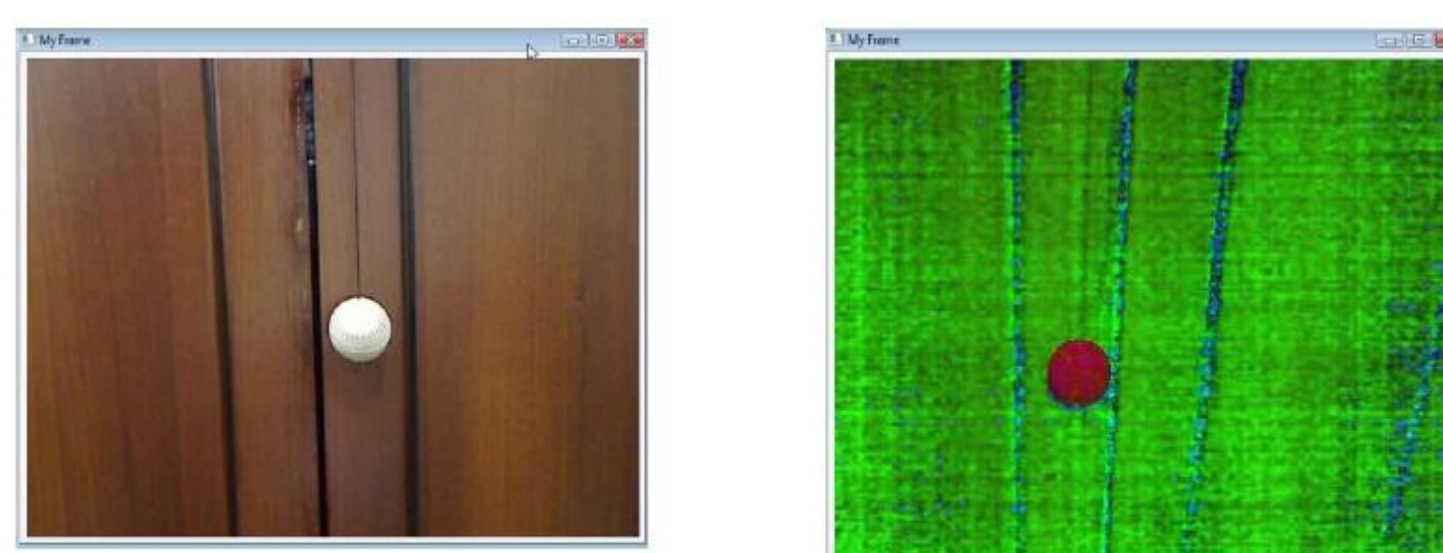
2.Synchronization : We used the send and receive communication operations to synchronize between master processor and slaves and vice verse.

Also a barrier is used to insure the correctness of the algorithm, we used a barrier in each slave after receiving objects and before feature extraction for the goal.

Algorithm:

Each processor uses PSO algorithm to track its object. We have four phases for each frame on each slave: Initialization, converting color model for frame, then particle initialization for each object in each slave see the following figures

After initialization, evaluate phase, predict phase in order to get the optimal or near the optimal solution, we calculate the best particle using fitness function.



a: RGB Image b: HSV Image

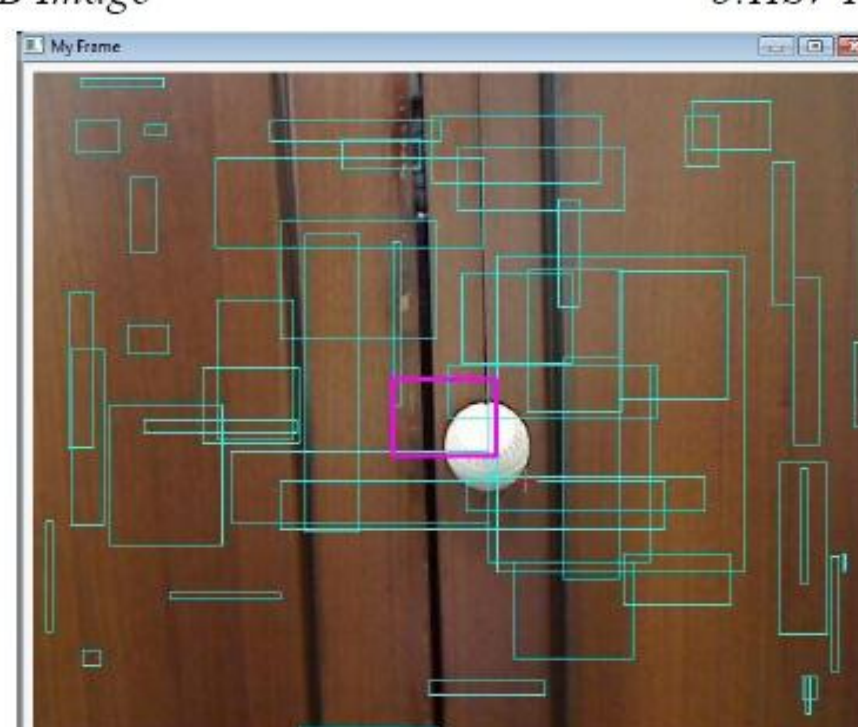


Figure 2: Initialization phase

Results and Experiments:

Testing Environment

We implement the system under ubuntu 11, the code was written on Message Passing Interface MPI, and this language enables to build parallel programs. The processors are Intel CoreTM i3-330M (2.13GHz, 4Threads, 3M cache) RAM: 2GB DDR3 @ 1066MHz.

Computational performance

The time for the master is 0.088, neglecting the taken time by the user while selecting the object.

Tracking performance

The average fitness value on each slave, we have seven iteration see figure 3

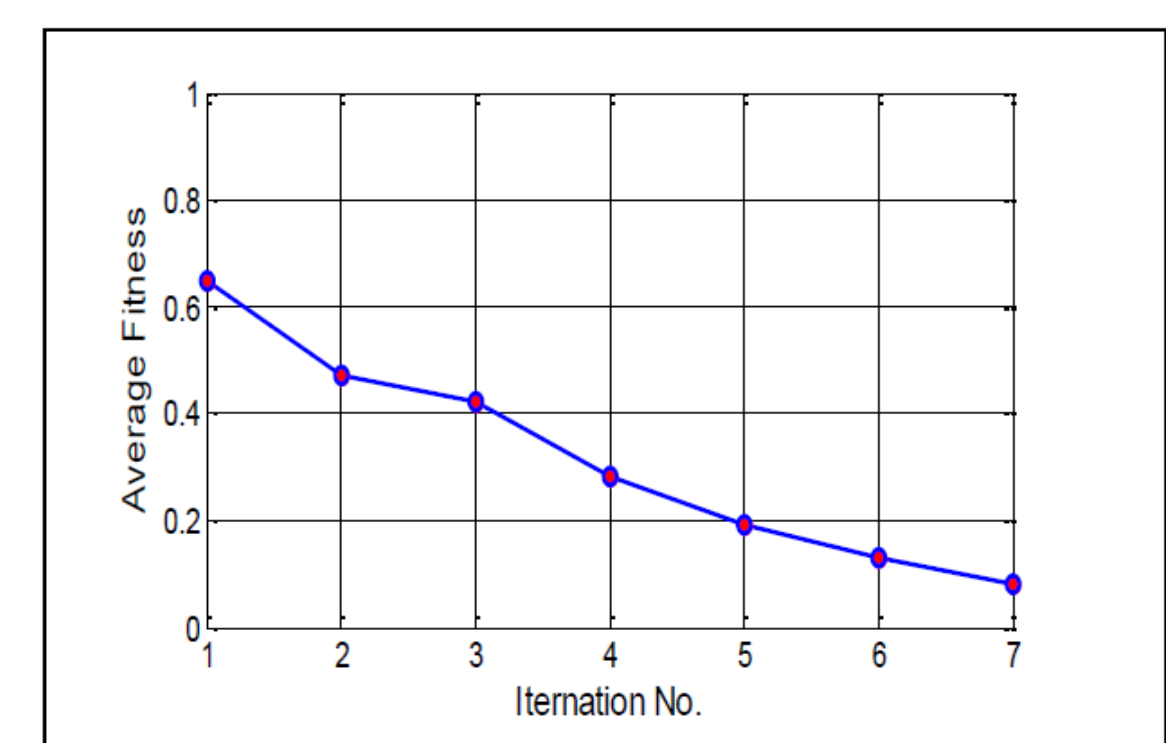


Figure 3 : Average fitness values in each iteration

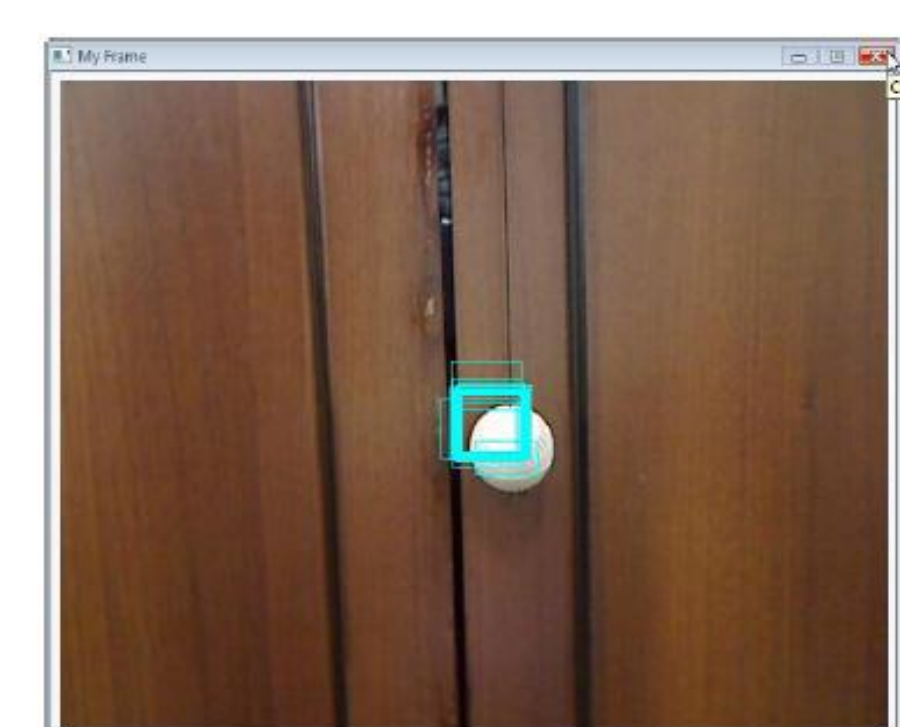


Figure 4: Convergence for particles to a given object

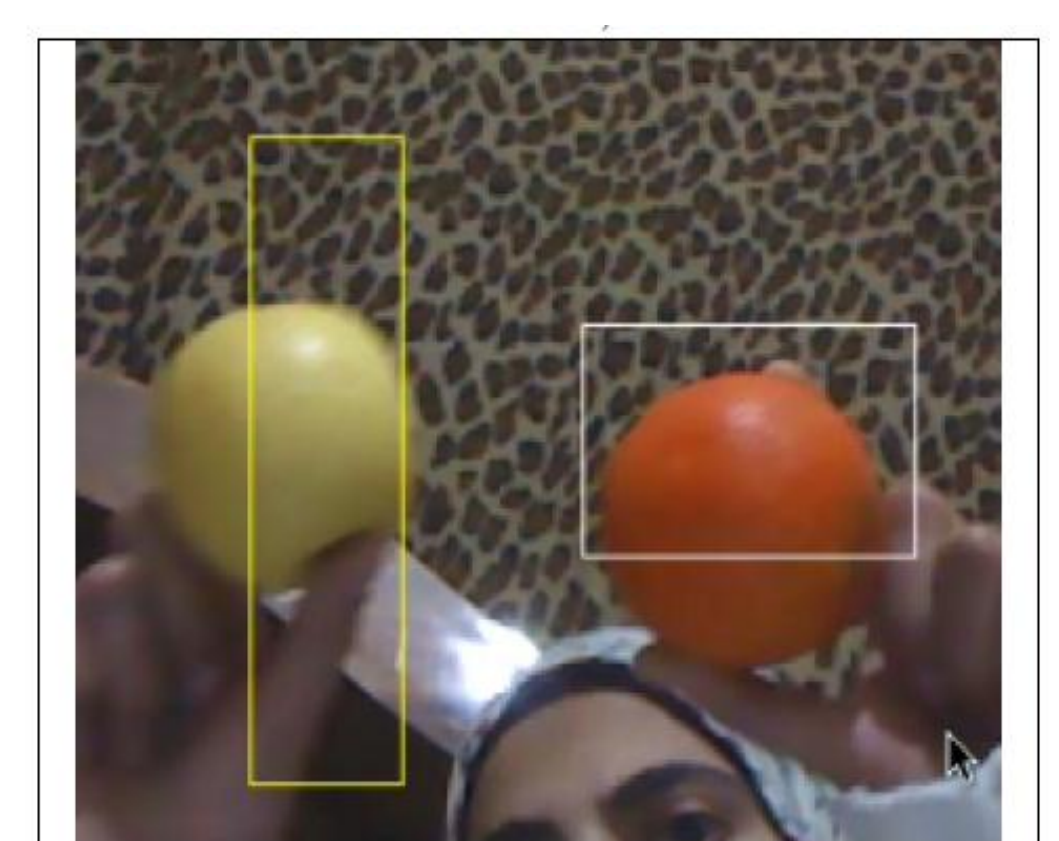


Figure 5: Tracking two objects, the result is shown on the master