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

Ahlam Hasan Albashiti
Master of Informatics
Palestine Polytechnic University
Hebron, Palestine
ahlam@student.ppu.edu

Mohammed ALDasht
Assistant professor at IT department
Palestine Polytechnic University
Hebron, Palestine
mohammed@ppu.edu

January 10, 2012

Abstract

This paper presents a parallel real time multiple objects tracking in dynamic environment, using Particle Swarm Optimization (PSO) algorithm. In our 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. The experimental results demonstrate the success of this system.

Keywords Multiple object tracking, Master-Slave parallel approach, PSO algorithm

1 Introduction

Object tracking means following and detecting a desired object in video sequences[1], multiple object tracking is to track multiple moving objects[2]. 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, the object occlusion etc. To enable tracking, we need an efficient algorithm that satisfies the three constrains. 1) It should be appropriate for real time cases. 2) It should be suitable under object transformation. 3) It should work under dynamic environment, and illumination changes.

PSO algorithm is an optimization algorithm that is inspired by the social behavior of a swarm of birds, fishes etc. to find the optimal solution in a search space [1]. To do this, each particle has a position and velocity in all dimensions. A fitness value is also maintained to evaluate the position for each particle [2][3].

This algorithm is suitable for real time application[4]; because of a set of individual behavioral rules that allow reaching the optimal or near the optimal solution quickly. Parameters of this algorithm are limited which makes it easy to optimize them.

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.

Outline This paper is organized as follows. Sect 2 studies related works, and sect 3 discusses the methodology, standard PSO, PSO-based tracking, and parallel multi object tracking, while sect 4 gives the results, finally the conclusion.

2 Previous work

There are many methods have been used for tracking such as mean shift algorithm[5] which is suitable in case of partial occlusion, but it is fail when the movement of the object is fast [6] , particle filter algorithm[7] is widely used but it suffers from the sample impoverishment problem[8]. It is proved that PSO algorithm in [9] is more efficient than particle filter.
Jilkov et al, (2011)[10]used a parallel particle filter algorithm for tracking multitarget, this algorithm is based on the Joint Multitarget Probability Density (JMPD) algorithm for multiple target tracking. The resample phase on the particle filter is parallelized. The results report the success of this method to track multitarget up to 50.
Thida et al,(2009) [6]build a multi-objects tracking in crowed scene using PSO. The fitness function for their system is defined using color based covariance matrix. Their experiments proved the success of PSO for tracking multiple objects.
there are several works for object tracking using PSO[11].

3 Proposed Method

1. **Standard PSO**
   PSO algorithm has three phases, initialization, optimization and termination. In initialization phase all particles are randomly initialized by applying the equation 1[12]. And the velocity is set to zero.

   $$x_i(t=0) = x_{min} + (x_{max} - x_{min}) * rand \quad (1)$$

   Optimization phase contains several iterations, in each iteration the positions and velocities of all particles are updated, in order to reach the optimal solution; this is done using the following velocity equation 4 and position equation 5[13]:

   $$local = w * v_i + c_1 * r_1 * (pBest_i(t) - x_i(t)) \quad (2)$$

   $$global = r_2 * (gBet_i(t) - x_i(t)) \quad (3)$$

   $$v_i(t+1) = local + global \quad (4)$$

   $$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

   Finally the termination phase, one way is reaching predefined number of iteration.

2. **PSO-based tracking**

   Particle shape
   Since we deal with frames (images) we represent each particle, in the same way as the goal, as a window (rectangle shape) with four dimensions, so each particle has center , width, and height. This window looks like the shape of the goal.

   Algorithm
   In our system we have four phases for each frame on each slave: Initialization, evaluate phase, predict phase, and resample phase. Initialization phase is done as same as standard PSO using equation 1 and the velocity is set to zero. On the other hand evaluation phase is used to evaluate the current position of each particle with respect to the goal. How far each particle from the goal? What is the best particle of the swarm (global best)? And what is the best position the particle has at the current frame (local best)? All these questions are answered after each evaluation phase.

   The three questions above are evaluated using the fitness function, the input of this function is a vector with four dimensions and the output is a value representing the distance of the particle's histogram from the goal's histogram which is calculated using the hue component of the HSV image.

The predict phase is done with approximately seven iterations, in each iteration all the particles are update their positions according to the equation 2 and 3 respectively in order to reach the best particle which is the nearest to the goal.

Instead of making initialization phase in each frame (which lead to lose the previous positions of the particles in the frame), we resample the particles in the search space for next frame so as to maintain new position near the previous one.

This is very important in the object tracking because the new poison of the moving object is assumed not far from its previous position. So if we re-initialize the particles we will lose this information that enables us to track the object.

3. **Parallel multiple objects tracking**
   The system is built using master-slave processers, the master distribute objects among slaves, each slave keep tracks its object using PSO algorithm and returns the position in each frame.
   Decomposition
   In our work we depend on domain decomposition, where 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.

   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 to prevent null array on calculating histograms. Also we used another barrier on the master before receiving the results from slaves to insure drawing particles on the same sent frame.
   Figure 1 shows the pseudo code of our parallel algorithm.

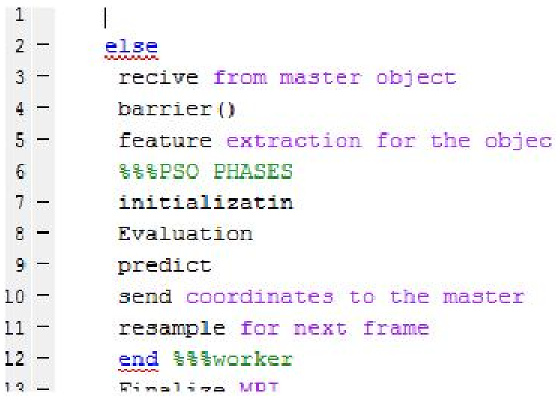
```
%% MASTER
if master then
  {  %%%%%Select OBJECTS
 while ( objectIsNotSelected)
    {   recive frames from webcam
        for (i=1;i<NumberOfProcessors;++i)
        {

            selectObject (i)
            send frame to processor (i)
            sendObject(processor i)
        }
        show frame
        press esc to exit
    }
    %%% TRACKING
 while( NOT objectIsNotSelected)
     {   recive frame from webcam
         send flag for all processor
         barrier()
        for (i=1;i<NumberOfProcessors;++i)
        {   color=0
            recive from processor(i)coordinates of particles
            drawParticles(coordinates,color)
            color=color+20
        }
        show frame
    }
   realese resourses
 }%%end master
```

Figure 1: The master pseud code

4 Results and Experiments

This section presents the testing environment in terms of software, hardware, and data, then obtained results is explained.
Testing Environment

We implement the system under ubuntu 11, the code was written on Massage 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.
We get the data in real time using the built in



```
else
 recive from master object
 barrier()
 feature extraction for the objec
 %%%PSO PHASES
 initializatin
 Evaluation
 predict
 send coordinates to the master
 resample for next frame
 end %%%worker
 Finalize MPI
```

Figure 2: The slave pseud code

webcam of the computer.
Computational performance
The time for the master is 0.088, neglecting the taken time by the user while selecting the object.
Tracking performance
Table 1 shows the results for our parallel algorithm. As we see, we test the parallel algorithm for two objects. The objects are different in their motion, color, shape.

Table 1: Frames from video sequences ,each object has different tracker)

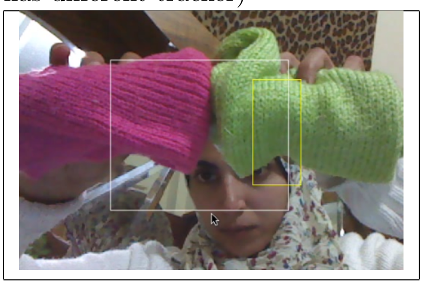


Table 2: Frames from video sequences ,each object has different tracker)

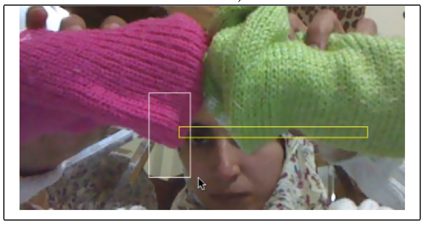
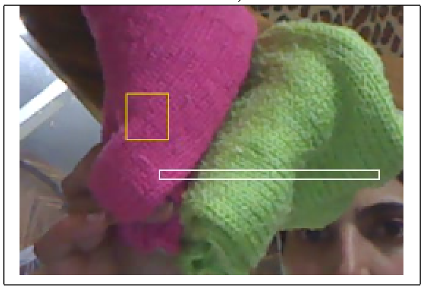


Table 3: Frames from video sequences ,each object has different tracker)



5 Conclusions

In this research, a new methodology for solving parallel real time general purpose multiple objects tracking is introduced using PSO algorithm. This system is built in master-slave processors, where the master processor distributes the objects among slaves, and all slaves track their objects and return results to the master. The advantages of this system are: it is a real time in dynamic environment, general purpose object regardless of the shape, color, motion, and the results showed good parallel multiple objects tracking algorithm.

References

[1] Kennedy.J and Eberhart.R, “Particle swarm optimization,” *IEEE*, 1995.

[2] Sulistijono.I and Kubota.N, “Human head tracking based on particle swarm optimization

Table 4: Frames from video sequences ,each object has different tracker)

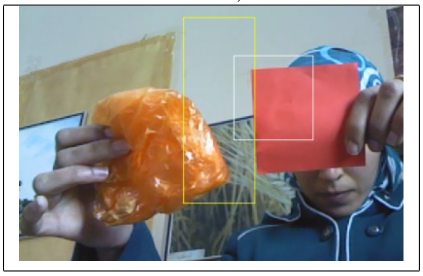
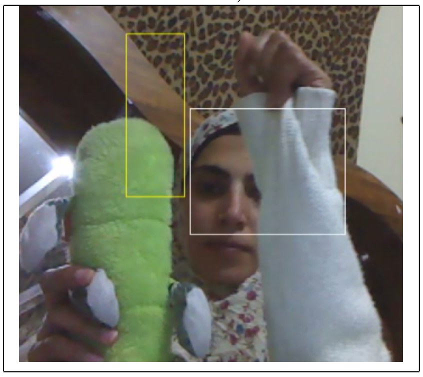


Table 5: Frames from video sequences ,each object has different tracker)



and genetic algorithm," *JACIII*, 2007.

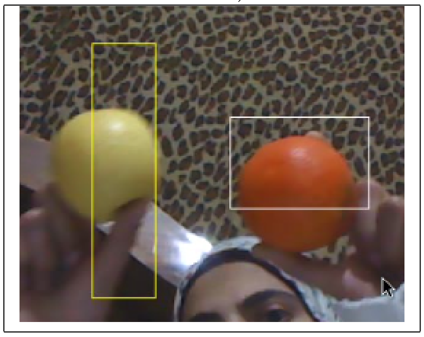
[3] Eberhart.R and Shi.Y, "Particle swarm optimization: Developments, applications, and resources," *IEEE*, 2001.

[4] Kobayashi.T, Nakagawa.K, and Zhai.G, "Real time object tracking on video image sequence using particle swarm optimization," *IEEE*, 2007.

[5] C. D and Ramesh.V, "Real-time tracking of non-rigid objects using mean shift," *IEEE.*, 2000.

[6] T. .C and R. .M, "A particle swarm optimization approach for multi-objects tracking in crowded scene," *12th International Conference onIEEE.*, 2009.

Table 6: Frames from video sequences ,each object has different tracker)



[7] N. .K, Koller-Meier, and Gool.L, "An adaptive color-based particle filter," *IEEE.*, 2002.

[8] Howida.A and Imbaby.A, "Particle filter versus particle swarm optimization for object-tracking," *Springer*, 2009.

[9] Zhang.X, Hu.w, and Zhu.M, "Sequential particle swarm optimization for visual tracking," *IEEE*, 2008.

[10] Jilkov.P and Wu.J, "Implementation and performance of a parallel multitarget tracking particle filter," *IEEE*, 2011.

[11] Musa.Z and Bakar.R, "Mobile camera tracking system using gbln-pso with an adaptive window," *Springer-Verlag*, 2011.

[12] Venter.G and Sobieski.J, "Particle swarm optimization," *IEEE*, 2002.

[13] Shi.Y and Eberhart.R, "Parameter selection in particle swarm optimization," *Springer-Verlag*, 1998.