Polytechnic Journal
Polytechnic Journal

Volume 9 | Issue 2 Article 31

12-1-2019

Particle Swarm Optimization on Parallel Computers for Improving the Performance of a Gait Recognition System

Shahla A. Abdulqader
Department of Computer Systems,Technical Institute, Mosul, Northern Technical University, Mosul,
shahlaak71@yahoo.com

Hasmek A. Krekorian
Department of Business Administration Techniques

How to Cite This Article

Abdulqader, Shahla A. and Krekorian, Hasmek A. (2019) "Particle Swarm Optimization on Parallel Computers for Improving the Performance of a Gait Recognition System," *Polytechnic Journal*: Vol. 9: Iss. 2, Article 31.
DOI: https://doi.org/10.25156/ptj.v9n2y2019.pp193-201

Particle Swarm Optimization on Parallel Computers for Improving the Performance of a Gait Recognition System

Abstract

In recent years, the gait recognition (GR) using particle swarm optimization (PSO) algorithm (OSO) has been execute very fast and accurate with single computer, but with the appearance of parallel computing (PC), it was necessary to use this technique to improve the results of GR. This study presents the use of parallel computing approaches (PCA) to implement PSO for a GR system (GRS) to decrease processing while maintaining reconstructed image quality. These approaches are: Codistributor and parallel cluster. Many experiments have been executed with recognition between the two approaches. The experimental results showed that increasing the PSO swarm size, decreasing number of iterations, and increasing number of workers used for the PCA can reduce recognition time and increase performance. Best results were obtained from implementing parallel computing with eight workers and 100 iterations. The execution time reached 4s and PSNR reached 44db. At the same time, the best results were obtained from PCL approach

Keywords

Codistributor, Discrete cosine transform, Gait recognition system, Parallel cluster, Parallel computing approaches

Polytechnic Journal. 2019. 9(2): 193-201
ISSN: 2313-5727
http://journals.epu.edu.iq/index.php/polytechnic

POLYTECHNIC JOURNAL

REVIEW ARTICLE

Particle Swarm Optimization on Parallel Computers for Improving the Performance of a Gait Recognition System

Shahla A. Abdulqader[1]*, Hasmek A. Krekorian[2]

[1]Department of Computer Systems, Technical Institute, Mosul, Northern Technical University, Mosul, Iraq, [2]Department of Business Administration Techniques, Technical College of Management, Northern Technical University, Mosul, Iraq

***Corresponding author:**
Shahla A. Abdulqader,
Department of Computer Systems, Technical Institute, Mosul, Northern Technical University, Mosul, Iraq.
Email: shahlaak71@yahoo.com

Received: 15 June 2019
Accepted: 03 September 2019
Published: 10 December 2019

DOI
10.25156/ptj.v9n2y2019.pp193-201

ABSTRACT

In recent years, the gait recognition (GR) using particle swarm optimization (PSO) algorithm (OSO) has been execute very fast and accurate with single computer, but with the appearance of parallel computing (PC), it was necessary to use this technique to improve the results of GR. This study presents the use of parallel computing approaches (PCA) to implement PSO for a GR system (GRS) to decrease processing while maintaining reconstructed image quality. These approaches are: Codistributor and parallel cluster. Many experiments have been executed with recognition between the two approaches. The experimental results showed that increasing the PSO swarm size, decreasing number of iterations, and increasing number of workers used for the PCA can reduce recognition time and increase performance. Best results were obtained from implementing parallel computing with eight workers and 100 iterations. The execution time reached 4_s and PSNR reached 44_{db}. At the same time, the best results were obtained from PCL approach.

Keywords: Codistributor; Discrete cosine transform; Gait recognition system; Parallel cluster; Parallel computing approaches; Particle swarm optimization

INTRODUCTION

Person identification can be associated with the features of any individual. Recently, its applications have been heavily researched, for different applications such as for authentication for computer systems (Sruthy, 2013) and using many techniques: Token, knowledge, and biometric based. A knowledge technique such as depends on a personal code (password) or a document and relies for identification on, say a passport, driver's license, or credit card. The disadvantages of these code may be stolen, lost, and forgotten. Biometrics use physiological features of a person and cannot be lost (Anil et al., 2004), for example, "fingerprint, iris, face, or speech recognition" (Gajanan et al., 2012). The requirements of this technique: (1) Controlled environment and (2) stand the subject at a fixed distance in front of the camera. Thus, it cannot be used in real time situations for people surveillance. Gait recognition (GR) provides a method to recognize people at a distance without their awareness. Recent research has focused on this approach (Sruthy, 2013).

Gait can be defined as a sequence of positions during walking that characterizes the individual. A single position can be identified within a typical motion sequence. It can be captured by two-dimensional (2D) video cameras of a typical monitoring system which stores video clips of the 2D images (Adam et al., 2011).

Thus, gait is seen as the coordinated, cyclic combination of movements that result in human locomotion. The movements are coordinated in the sense that they must occur with a specific temporal pattern for the gait to occur. The movements in a gait repeat as a walker cycles between steps with alternating feet. It is both the coordinated and cyclic nature of the motion that makes gait a unique phenomenon. Examples of motion can be any activity a person carries out such as walking, or climbing. GR relies on the reliable identification of individual characteristics within the ambulatory sequence (Massimo et al., 2005). Recently, there has been an increased interest in visual surveillance systems for human identification from a distance based on the extraction of movement features for recognition. Visual development in the computing had a clear effect in making the automatic analysis of walking possible. For this reason, studies have been directed to combining human motion analysis and biometrics in surveillance systems depending on distance. This concern came from the need of monitoring system for security-sensitive sites such as banks and airports (Bogdan et al., 2014).

Particle swarm optimization (PSO) is a heuristic, population-based, and self-adaptive search optimization technique that is based on swarm intelligence to solve problems in many applications. It comes from the research on the bird movement behavior. Its algorithm is used and developed for its simplicity in implementation (Qinghai, 2010). PSO was first introduced in "1995 by Kennedy and Eberhart" (James and Russel, 1995). Many researches focused on developing and enhancing the PSO (Voratas, 2012); and (Peyvandi et al., 2011) used in solving identification problems such as face recognition (Shinde et al., 2012) and palm print recognition (Arunkumar and Valarmathy, 2013); and Ola et al., 2013. Other researchers used GR that are based on PSO (Spela et al., 2008) to recognize different movement of the body, while others use PSO in Neural Networks and Fuzzy Logic for GR (Qiang et al., 2018). It is also used in detecting diseases related to gait such as Parkinson's (Xu et al., 2018). Discrete cosine transform (DCT) had been introduced by Nasir et al., 1974, and it is used widely in image processing, therefore many researcher used in recognition process to extract features Shahlla and Omaima, 2015; and Zheyi et al., 2016.

A number of studies have used PSO for implementing gait analysis to minimize recognition time. Most of them have noise ratio problem and large computational time, whereas few studies used parallel computing (PC) to reduce computation time. Therefore, in this research, parallel computing (PC) was chosen to implement PSO for GR.

PC uses control programming environment by sending programs to connected computers without the difficulty of logging in, running program, take results, and transferring files (MATLAB Parallel Computing, R2014b); and MATLAB Optimization, R2014b. It is possible to solve computational and data problems using multiprocessor computers and by distributing computing server software in MATLAB. It includes: Parallel for-loops (ParFor), distributed arrays, parallel numerical algorithms, and message-passing functions to implement (task and data) parallel at high level without programming for specific hardware and network architectures (MATLAB Parallel Computing, R2014b). For that, two PC approaches were discussed and executed "CoDistributor" (COD) and "Parallel Cluster" (PCL). Parallel PSO (PaPSO) system for GR used to improve performance suggested system that use DCT and PSO in PC. This paper contain: Section 2: Description of parallel GR (PGR) system, Section 3: Research methodology, Section 4: Results, and finally Section 5: The conclusions.

PGR SYSTEM

The strategy of parallel system for GR includes the flowing:

PSO Algorithm

PSO can be implemented easily on a large number of samples. It includes the following steps:

1. Each solution which represents an individual of a population is implemented as a particle N-dimension
2. Each particle adjusts its position, move closer to the optimal point and evaluates their position. Knowing that each particle associated with the fitness function
3. Particles compare themselves to their neighbors and imitate the best of them
4. Pbest: Represents the best value of the particle I
5. Gbest: Best value that one of the swarm particle reaches
6. Lbest: Best value that particle in a local swarm reaches.

$$V_i(i+1)=W*V_i(t)+C1*rand*(Pbest(t)-X_i(t)+Ct)-X_i(t)+C2*rand*(Gbest(t)-X_i(t)) \quad (1)$$

Where: X_i: i^{th} particle of swarm, $V_i(t)$: Particle velocity,

W: Weight (random No. between 0, 1), (C1, C2): Speeding factors (with value 2).

This equation is used to compute the new velocity of each particle, and from it, the new velocity $V_i(t+1)$ is computed, which is affected by:

Pbest, Gbest and $V_i(t)$: Velocity of i^{th} particle X_i in time t.

This equation used to compute new fitness value of each particle:

$$X_i(t+1)=X_i(t)+V_i(t+1) \quad (2)$$

It will change its value according to its new velocity($V_i(t+1)$). PSO algorithm was described in researches (Qinghai, 2010); (James and Russel, 1995); (Voratas, 2012); and (Peyvandi et al., 2011):

1. Initialize parameters (number of generations, population size, weights, c1, and c2)
2. Initialize population (velocity and position of each particle) and initialize Pbest and Gbest.
3. New generation
4. Take one particle (P) from population
5. Compute: (new velocity [Pvelocity] of particle using Eq.(1), new position (Pposition) of particle using Eq.(2)
6. Pbest = Pposition if cost (Pposition ≤ cost (Pbest)
7. Gbest = Pbest if cost (Pbest) ≤ cost (Gbest)
8. Repeat steps (4.9) until there are no more particles in the population

9. Repeat steps (3.9) until reaching maximum number of generations
10. Return Gbest.

DCT

DCT transforms the input image into a linear combination of weighted basis functions, transform image from spatial domain to a frequency domain. It uses cosine base functions, exhibits good de-correlation, and energy compaction characteristics. It is help to separate important parts of the visual image. This information occurs at low frequencies while the high frequencies of the image are very small and can be removed as distortion. For this reason, it is used as a feature reduction and extraction method by many researchers and to improve recognition rate (Nasir et al., 1974).

Parallel Computing Approaches

PC includes the exchange of information between many connected computers to increase the speed of computations and provide large amounts of memory for program execution (Hahn et al., 2009). PC is an extension of MATLAB that takes advantage of multicore desktop and clusters. It runs on a desktop and can take specifications of up to 8 cores and parallel programs can be run interactively or in batch.

A distributed computing server controls the parallel execution on a cluster with tens or hundreds of cores. There are several ways to execute the parallel program: Interactive local (matlabpool); and indirect (local and remote), (batch) (MATLAB Parallel Computing, R2014b). Message Passing Interface (MPI) is a standard method to implement parallel programs on many processors in PC. A set of Matlab scripts (MatlabMPI) is used to support built-in data types and data structures that supported in the parallel programming environment (Message-Passing Interface, 2008); and (Jeremy, 2001).

Codistributor

The arrays of codistributed (COD) are special arrays to store segments of data on MATLAB participate in PC problem. It can treat larger data sets and be structured in several ways: Using functions such as rand, ones, and zeros and connecting arrays in different data in different labs with the same name. The COD arrays are partitioned into segments and each one stored in the workspace of different labs, so reducing the size of array and process, which mean faster processing and efficiency in using the memory. Within the parallel pool using COD, it is possible to access data of arrays distributed among workers (w) and use one management process to supervise workers who execute on a single program. Each worker has an identifier; knows how many other workers there are; determines their behavior based on ID, runs on a separate core and there is a separate workspace used for each worker (MATLAB Parallel Computing, R2014b). In this study, the swarm size is distributed to workers to improve the speed of processing.

Parallel cluster

PC software has the ability to run a local cluster of workers on client machines to run jobs without requiring distributed computing server software. For that, all processing required for client, scheduling, and task evaluation is performed on the same computer. This gives the opportunity to improve, test, and debug parallel applications before running them on the network cluster. The functions used in cluster profiles are: Batch; parpool; and parcluster. Any job on the local cluster can be programmed when it is necessary need to determine the process for evaluating functions using PC. The basic steps in creating and running a job that contains simple tasks are (MATLAB Parallel Computing, R2014b):

1. Set a cluster by using parallel.defaultClusterProfile and parcluster to create object c, parallel. Default cluster profile (“local”), c= parcluster ()
2. Set job j on the cluster by executing this line: j = createJob (c)
3. Set tasks within job j. Each task evaluates mathematical operations that are passed as inputs argument “createTask (j,@operation,1,{[1,1]})…to{[n n]}.”
4. Input job to queue for running. The scheduler distributes the job’s tasks to workers. The local cluster starts worker sessions using: submit (j)
5. Wait; for job to complete, then get all of the results from tasks of the job
6. Output the result then; remove job from scheduler’s storage location.

RESEARCH METHODOLOGY

The methodology depends on a database taken from CASIA (CASIA Gait Database, 2006), the data were collected for each person’s silhouette in height and width with different views. Each person has sequences of 20 images, four sequences and three directions (parallel 0, 45°, and 90° to image plane). The length of each sequence will not be identical due to the variation of the walker’s speed. PGR system based on PSO and DCT for features extraction is suggested in this work using MATLAB, 2015. The DataBase includes images for 20 persons, each person with 50 images (size of 240×352 pixels), four cases for three angles. The final database includes: 20 person × 3 × 4 × 50 = 12,000 images.

Furthermore, in this study two parallel approaches from MATLAB PC such as COD and PCL were used to implement PSO for GR. The execution according to Manager/Worker model of parallel programming to reduce computation time and increase recognition performance will be calculated (PSNR is an efficient equation for evaluating the accuracy of the processed images and MSE is considered one of the measurements used in comparing the accuracy of the results from the processing of the images) using Eq.(3) and Eq.(4). Nine computers were used (1 Manager and 8 Workers). The execution environment of MATLAB requires configuration before carrying out the parallel programming. This procedure has been introduced in MATLAB toolbox user guide. MATLAB distributed computing server (MDCS) (MATLAB: Distributed Computing Server, 2015) is adopted for the server and each worker. This MDCS is downloaded and installed on all computers used to establish the parallel programming service. Manager/worker technique was selected from MDCS. PSO parameters can be set by manager as follows: 40, 32 Swarm size, 250,150,100 number of iterations for the main computer (Manager) Table 1, and two other values (5 or 4) are used as Swarm size per worker. The suggested PCA for GR based PSO is summarized in Figure 1.

$$MSE = \frac{\sum_{i=0}^{M}\sum_{j=0}^{N}\left|X(i,j)-X_c(i,j)\right|^2}{M\times N} \qquad (3)$$

$$PSNR = 10Log_{10}(\frac{255^2}{MSE}) \qquad (4)$$

Table 1: Parameters of parallel PSO

Swarm size N	40,32
C1, C2	2
Weight	0.5
Number of Iterations	250,150,100
Block of image	70×70
Times	Ts
Workers	W

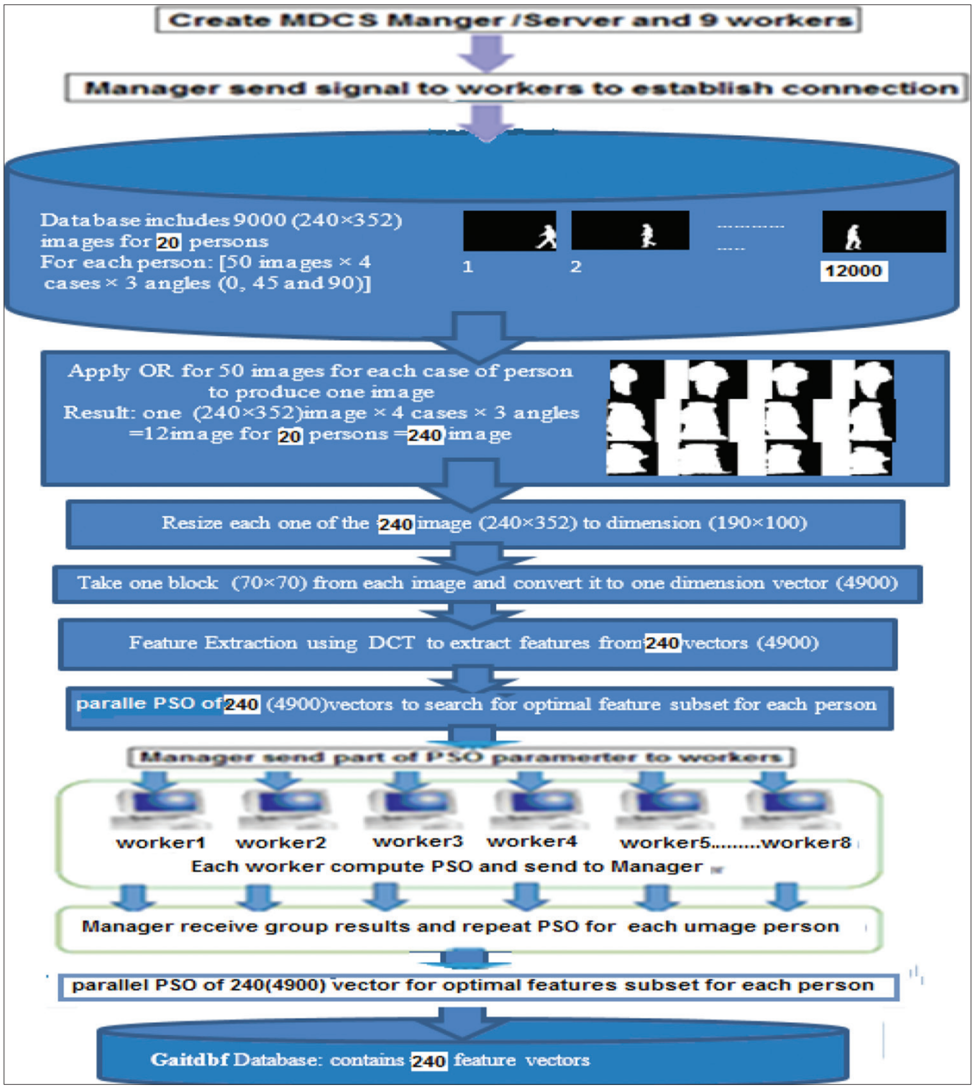
Create MDCS Manger /Server and 9 workers
Manager send signal to workers to establish connection
Database includes 9000 (240×352) images for 20 persons For each person: [50 images × 4 cases × 3 angles (0, 45 and 90)]
1
2
12000
Apply OR for 50 images for each case of person to produce one image Result: one (240×352)image × 4 cases × 3 angles =12image for 20 persons =240 image
Resize each one of the 240 image (240×352) to dimension (190×100)
Take one block (70×70) from each image and convert it to one dimension vector (4900)
Feature Extraction using DCT to extract features from 240 vectors (4900)
paralle PSO of 240 (4900)vectors to search for optimal feature subset for each person
Manager send part of PSO paramerter to workers
worker1
worker2
worker3
worker4
worker5.........worker8
Each worker compute PSO and send to Manager
Manager receive group results and repeat PSO for each umage person
parallel PSO of 240(4900) vector for optimal features subset for each person
Gaitdbf Database: contains 240 feature vectors

Figure 1: Parallel gait recognition system

Processing of PGR

The processing of the PGR system contains the following steps is shown in Figure 1:

1. Create MDCS manager/server and eight workers
2. Establish connection to send signal from manager to workers
3. Input: 50 images size (240,352) pixels, 4 states, and 3 angles (0°, 45°, and 90°) for any image

   The OR logical gate will be applied on every 50 images to produce only one average image for each four case and for each angle. Then, the total number of images resulted from this process are:($20 \times 3 \times 4 \times 1 = 240$) images for 20 persons and resize each one of images from size 240×352 pixels to size 190×100 pixels to reconstruct the extraction rectangle to include the person without extra black pixels and obtain height and width of the person; the sequence is calculated and each frame is converted to biggest height and width.
4. Extract block (70×70) from each image (190×100) pixels
5. Transform each sub image block from a 2D array to 1D array
6. The feature of the person will be extracted by applying the DCT algorithm
7. Parallel PSO for classification is used for each one of the 240 feature vectors to execute for eight workers as follows in one of the two ways (COD and PCL):
   1. Set PSO parameters as shown in Table 1 then distributed data and send to the workers
   2. Set position, velocity, "Pbest" and "Gbest." Then, Compute "fitness function" in each worker
   3. Compute optimal value of "Pbest" and collection "Gbest" according
   4. Compute new speed and new position of practical using (1) and (2)
   5. Send result to manager
   6. Repeat steps (4–6) end of the iterations
   7. Save subset features in database: "Gaitdbf."

Test processing of PGR

The steps of PGR system are described with details in Figure 2.

EXPERIMENTAL RESULTS

Parallel computers were used for PSO using MATHLAB 2015 and many experiments were applied to the system. The database includes 12,000 images size (240×352) pixels for 20 persons was selected from "CASIA Database" (CASIA Gait Database, 2006). Images were obtained by taking 3 angles: 0°, 45°, and 90°, with four cases for each angle and 50 images for each case, then images were re-sized to 190×100 pixels. The suggested system performance was compute by PSNR and MSE. The first three experiments were done by setting PSO swarm size equal 40 and iterations 250, 150, and 100. Each experiment included the two methods (COD, PCL). Table 2 shows the results (PSNR, MSE, and computation time) for the first experiment using parameters (swarm size equal 40, 250 iterations, and same number of workers). This demonstrates a decrease in computation time when we increased the number of workers. Another two experiments were accomplished by setting PSO swarm size equal 40 and iterations were 150 and 100. Experiments 2 and 3 used the same methodology as the pervious experiment. Table 2 and Table 3 illustrate the result (PSNR, MSE, and computation time) for these experiments recognition time with increasing number of workers.

At the same time, Table 2-4 demonstrate that with the same swarm size, the computation time is decreased when the number of iterations is decreased. From Table 2, the execution speed of GR by PCL has reached 65_s with PSNR equal 34_{db} when conducting one worker with swarm size equal 40, whereas the time required by COD is 72_s. Increasing the number of workers from 2 to 8 leads to a decrease in the time of execution to 6_s with using

Table 2: PGR (COD, PCL when swarm size=40 and iteration=250)

Swarm size=40, iteration=250						
W	COD			PCL		
	T_s	$PSNR_{db}$	MSE	T_s	$PSNR_{dn}$	MSE
1	72	34	0.0071	65	34	0.0069
2	66	33	0.0067	50	32	0.0061
3	60	33	0.0066	50	33	0.0059
4	51	34	0.0059	42	34	0.0051
5	32	35	0.0057	25	36	0.0042
6	22	35	0.0055	15	37	0.0040
7	17	36	0.0049	11	38	0.0034
8	10	38	0.0039	6	40	0.0026

Table 3: PGR (COD, PCL when swarm size=40 and iteration=150)

Swarm size=40, iteration =150						
W	COD			PCL		
	T_s	$PSNR_{db}$	MSE	T_s	$PSNR_{db}$	MSE
1	70	35	0.0065	60	36	0.0061
2	61	34	0.0062	45	35	0.0055
3	55	35	0.0061	41	34	0.0051
4	45	35	0.0049	39	36	0.0045
5	30	36	0.0052	22	37	0.0038
6	21	37	0.0050	14	38	0.0036
7	15	37	0.0041	10	38	0.0030
8	9	39	0.0022	5	42	0.0020

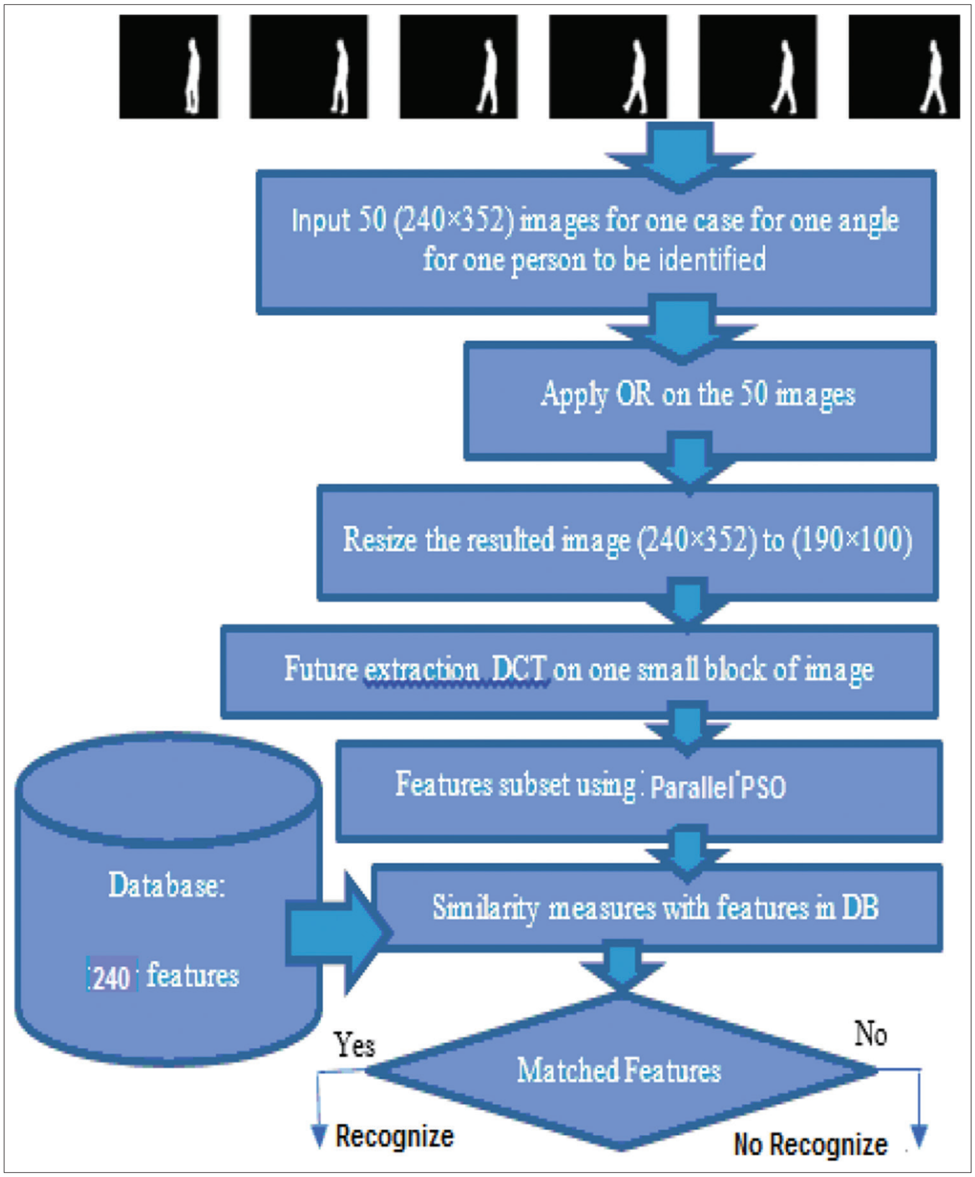
Input 50 (240×352) images for one case for one angle for one person to be identified
Apply OR on the 50 images
Resize the resulted image (240×352) to (190×100)
Future extraction DCT on one small block of image
Features subset using Parallel PSO
Database:
240 features
Similarity measures with features in DB
Matched Features
Yes
No
Recognize
No Recognize

Figure 2: Test processing of parallel gait recognition system

eight workers with improving in PSNR which reach 40_{db} and MSE equal 0.0026 in PCL, whereas execution time reaches 10_s in eight workers, PSNR equal 38_{db} and MSE equal 0.0039 with COD way with the same parameters. This means that it can be considered PCL is better than COD.

Table 4: PGR (COD, PCL when swarm size=40 and iteration=100)

W	Swarm size=40, iteration =100					
	COD			PCL		
	T_s	$PSNR_{db}$	MSE	T_s	$PSNR_{db}$	MSE
1	67	36	0.0060	55	37	0.0055
2	60	36	0.0055	40	38	0.0049
3	51	37	0.0057	39	38	0.0041
4	40	37	0.0047	30	38	0.0041
5	28	38	0.0050	21	39	0.0032
6	18	38	0.0047	13	40	0.0031
7	11	38	0.0039	9	42	0.0027
8	9	39	0.0025	4	44	0.0018

Tables 3 and 4 demonstrate that execution time is reduced by increasing the workers in the two ways by keeping swarm size equal 40 and iteration decreases from 150 to 100. Whereas the execution time reach 4_s in eight workers with increased value of PSNR to 44_{db} and the value of MSE was decreased to 0.0018 using PCL ways and the number of iteration equal 100. While the execution time reach 9_s, PSNR equal 39_{db} and MSE equal 0.0025 with same numbers of iterations and workers in COD way. This means that accuracy and execution speed using PCL compared with COD for GR with greater accuracy, as shown in Figures 3 and 4.

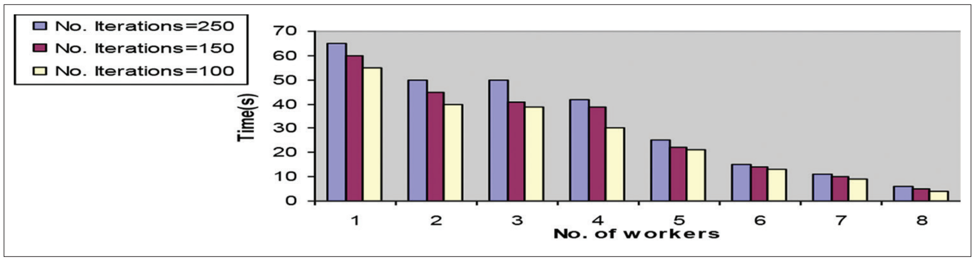

Figure 3: Time required for parallel gait recognition system using COD (swarm size=40)

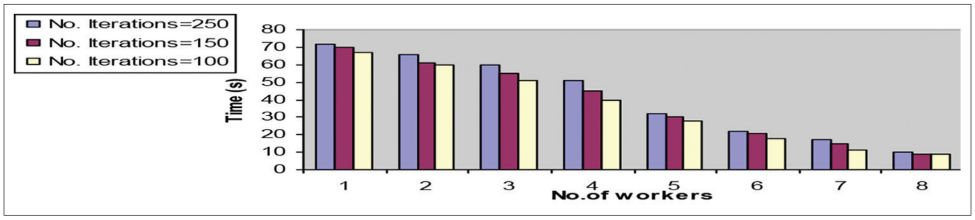

Figure 4: Time required for parallel gait recognition system using PCL (swarm size=40)

To test the accuracy of the results, the experiment was repeated for swarm size equal to 32 and with the same number of iterations used in the previous experiments (100, 150, and 250). The results are shown in Tables 5-7 that the time of implementation was also decreased with increasing in the workers, it reaches the lowest time of implementation with eight workers in the two ways. Table 7 shows that the implementation time using the PCL method is equal to 10_s which is less than the implementation time when using COD with same number of workers, while the value of PSNR was reduced compared with the previous experiments. It reached 35_{db} using COD with less error equal to 0.0035, while it reached 40_{db} in the PCL method with less error that reached 0.0023 with the same number of workers equal to 8 and iterations 100 and at swarm size equal 32, as shown in Figures 5 and 6.

It can be said that from all the previous experiences of both PCL and COD ways that the PCL method is better, where the value of swarm size equal 32, number iterations 100 and increased number of workers which increase GR with shorter execution time in all cases**,** wher**e** the value of PSNR that reaches 40_{db} with an execution time 10_s and eight workers. On the other hand, that increasing the workers and the value of swarm size to 40 reduced the execution time which reaches 4_s with increase PSNR to 44_{db} for the same number of iterations Figure 7. The results indicate that with increasing the size of the data, computers (workers) with a high capacity. PC is connecting, dividing the data and processors lead to speed up the executions. This itself speed up for any algorithm or method and applications for any amount of data. With the development of computers and communication devices, PC becomes easier, faster, and more efficient to implement any application and algorithm.

Table 5: PGR (COD, PCL when swarm size=32 and iteration=250)

Swarm size=32, iteration =250						
W	COD			PCL		
	T_s	$PSNR_{db}$	MSE	T_s	$PSNR_{db}$	MSE
1	74	33	0.0071	65	32	0.0069
2	67	31	0.0067	55	30	0.0061
3	63	31	0.0066	52	31	0.0059
4	55	32	0.0059	45	33	0.0051
5	44	32	0.0057	33	34	0.0042
6	33	33	0.0055	25	35	0.0040
7	21	33	0.0049	19	36	0.0034
8	17	35	0.0039	15	36	0.0026

Table 6: PGR (COD, PCL when swarm size=32 and iteration=150)

Swarm size=32, iteration =150						
W	COD			PCL		
	T_s	$PSNR_{db}$	MSE	T_s	$PSNR_{db}$	MSE
1	74	34	0.0068	62	35	0.0059
2	65	32	0.0065	51	36	0.0051
3	58	32	0.0065	48	35	0.0049
4	49	33	0.0055	43	37	0.0040
5	39	32	0.0051	30	38	0.0035
6	28	34	0.0050	22	39	0.0034
7	25	34	0.0043	15	39	0.0029
8	16	35	0.0033	13	39	0.0018

The research conducted by Shahlla and Omaima, 2015 and has some similarities to this study but also many differences. Table 8 shows the improvement of features clear by the value of PSNR with increasing the number of computers (workers), also decreasing the execution time and MSE in comparing with the aforementioned research.

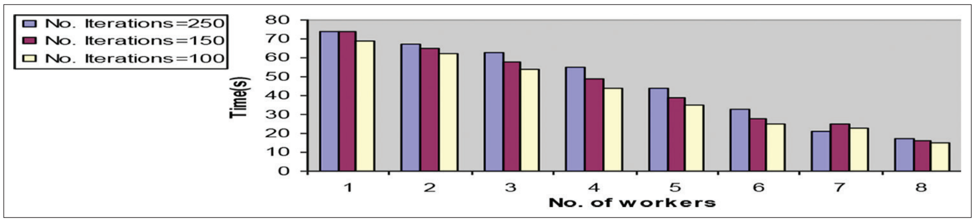

Figure 5: Time required for parallel gait recognition system using COD (swarm size=32)

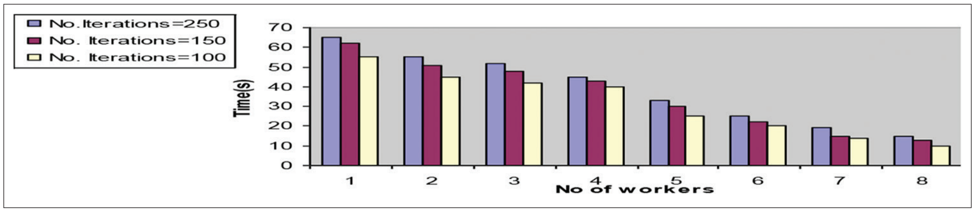

Figure 6: Time required for parallel gait recognition system using PCL (swarm size=32)

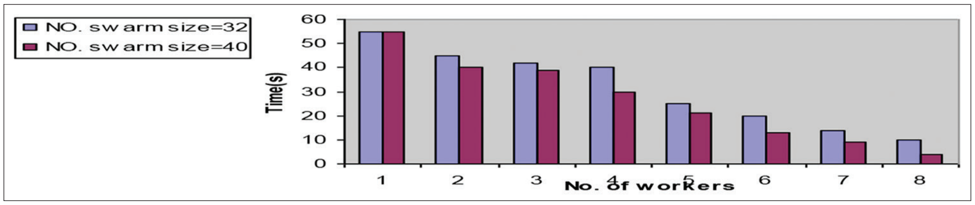

Figure 7: Time required for parallel gait recognition system using PCL (no. iteration=100)

Table 7: PGR (COD, PCL when swarm size=32 and iteration=100)

Swarm size=32, iteration =100						
W	COD			PCL		
	T_s	PSNR	$PSNR_{db}$	T_s	$PSNR_{db}$	MSE
1	69	35	0.0068	55	36	0.0055
2	62	33	0.0065	45	3	0.0048
3	54	33	0.0065	42	35	0.0045
4	44	34	0.0055	40	38	0.0039
5	35	34	0.0051	25	38	0.0031
6	25	35	0.0050	20	40	0.0030
7	23	35	0.0043	14	40	0.0025
8	15	35	0.0033	10	40	0.0023

Table 8: Comparisons with reference (Shahlla and Omaima, 2015)

	This study	(Shahlla and Omaima, 2015)
Swarm size N	Swarm size N=40, 32	Swarm size N=40,30,20
Number of iterations	Number of Iterations=250, 150, 100	Number of iterations=150, 100
Block of image	Block of image=70×70	Used many block of image
No. of computers	Used (9) computers	Used one computer
PSNR	PSNR reached 44 db	PSNR reached 35_{db}
MSE	MSE reached 0.0018	MSE reached 0.0088
Time	Time reached=4s	Time reached=67s, 55s

CONCLUSION

In this research, the computation time is decreased by two parallel ways (COD and PCL) for GR based PSO. Many variations were conducted using: 20 subjects; different swarm size (40 and 32); controlling PSO parameters. This improved PSNR and MSE while decreasing the computation time and increasing the number of workers in parallel programming required for GR based PSO. Furthermore, this time could be decreased with the increase of the swarm size from 32 to 40 and decreasing the number of iterations to equal 100. The experimental results showed that the PCL requires less computation time than COD. At the same time, the best quality of image (PSNR) can be increased when adopting PCL with less error. The results of "Parallel Cluster" are better than the results of the "CoDistributor" method.

REFERENCES

Adam, Ś., P. Andrzej and W. Konrad. 2011. Human Identification Based on the Reduced Kinematic Data of the Gait. 7th International Symposium Image and Signal Processing and Analysis, IEEE. p650-655.

Anil, K. J., R. Arun and P. Salil. 2004. An introduction to biometric recognition. IEEE Trans. Circuits Syst. Video Technol. 14(1): 4-19.

Arunkumar, M. and S. Valarmathy. 2013. Palmprint and face based multimodal recognition using PSO dependent feature level fusion. J. Theor. Appl. Inf. Technol. 57(3): 337-346.

Bogdan, K., K. Tomasz, M. Agnieszka and J. Henryk. 2014. 3D Gait recognition using spatio-temporal motion descriptors. In: Intelligent Information and Database Systems. Springer International Publishing, Berlin, Germany. p595-604.

CASIA Gait Database. 2006. CASIA Gait Database Collected by Institute of Automation, Chinese Academy of Sciences and a Citation to CASIA Gait Database. Available from: http://www.sinobiometrics.com. [Last accessed on 2019 May 20].

Gajanan, P. K., L. P. Satish and C. J. Dinesh. 2012. Human computer interpreting with biometric recognition system. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2(12): 140-147.

Hahn, K., M. Julia and K. Jeremy. 2009. Introduction to Parallel Programming and pMatlabv2.0. MPI: A Message-Passing Interface Standard, 2008, Version 2.1, University of Tennessee, Knoxville, Tennessee. Available from: http://www.mpi-forum.org/docs/mpi21-report.pdf. [Last accessed on 2019 May 20].

James, K and E. Russel. 1995. Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, WA. p1942-1948.

Jeremy, K. 2001. Parallel Programming with MatlabMPI, In: 5th High Performance Embedded Computing (HPEC2001) Workshop, MIT Lincoln Laboratory, Lexington, MA.

Massimo, T., B. Josef and G. Enrico. 2005. Biometric Gait Recognition. Biometrics School, United States. p19-42.

MATLAB Parallel Computing Toolbox User's Guide 4.3, R2014b.

MATLAB, Optimization Toolbox™ User's Guide, R2014b.

MATLAB. 2008. Programming Fundamentals. The MathWorks™, Inc. http://www.mathworks.com. [Last accessed on 2019 May 20].

MATLAB. 2015. Distributed Computing Server™ 5 System Administrator's Guide. The MathWorks, Inc., Apple Hill Drive, Natick, MA, Math Works, Inc. http://www.mathworks.com. [Last accessed on 2019 May 20].

Nasir, U. A., T. Natarajan and K. R. Rao. 1974. Discrete cosine transform. IEEE Trans. Comput. 23: 90-94.

Ola, M., A. Tarek, I. Gouda and M. O. Hoda. 2013. An adaptive multimodal biometrics system using PSO. Int. J. Adv. Comput. Sci. Appl. 4(7): 158-165.

Peyvandi, M., M. Zafarani and E. Nasr. 2011. Comparison of particle swarm optimization and the genetic algorithm in the improvement of power system stability by an SSSC-based controller. J. Electr. Eng. Technol. 6(2): 182-191.

Qiang, Y., X. Yi and Y. Zhiming. 2018. Classification of gait pattern in patients with neurodegenerative disease using adaptive neuro-fuzzy inference system. Comput. Math. Methods Med. 2018: 9831252.

Qinghai, B. 2010. Analysis of PSO algorithm. Comput. Inf. Sci. 3(1): 180.

Shahlla, A. and A. Omaima. 2015. Particle Swarm Optimization Based Discrete Cosine Transform for Person Identification by Gait Recognition. ICIT 2015 the 7th International Conference on Information Technology.

Shinde, P. V., L. G. Baisa and G. Rameshwar. 2012. Face Recognition Using Particle Swarm Optimization. Emerging Trends in Computer Science and Information Technology. p11-13.

Spela, I., T. Emanuele and R. P. Yvan. 2008. Human body pose estimation with particle swarm optimization. Evol. Comput. 16(4): 509-528.

Sruthy, S. 2013. Activity based person identification using particle swarm optimization algorithm. Int. J. Comput. Sci. Mob. Comput. 2(7): 1-6.

Voratas, K. 2012. Comparison of three evolutionary algorithms: GA, PSO, and DE. Ind. Eng. Manag. Syst. 11(3): 215-223.

Xu, C., T. Chen, Y. Xiaohui and S. Yining. 2018. Detecting Parkinson's disease using gait analysis with PSO. In: Human Aspects of IT for the Aged Population. Applications in Health, Assistance, and Entertainment. Springer, Berlin, Germany. p263-275.

Zheyi, F., J. Jiao, W. Shuin and H. Zhonghang. 2016. Human Gait Recognition Based on Discrete Cosine Transform and Linear Discriminant Analysis, IEEE International Conference on Signal Processing, Communications and Computing.