

Three variants Particle Swarm Optimization technique for optimal cameras network two dimensions placement

Salim Abdesselam[✉], Zine-Eddine Baarir

LESIA Laboratory, Department of electrical engineering, Biskra University, P.O.BOX 14 RP, Biskra, Algeria

Received 23 April 2017

Published online: 24 May 2018

Keywords

Standard Particle Swarm Optimization

Weight Particle Swarm Optimization

Canonical Particle Swarm Optimization

Motion Capture

Optimal two-dimensional placement

Abstract: This paper addresses the problem of optimal placement in two-dimensions of the cameras network for the motion capture (MoCap) system. In fact, the MoCap system is a three-dimensional representation environment used mainly to reconstruct a real motion by using a number of fixed cameras (in position and pose). The main objective is to find the optimal placement of all cameras in a minimal time under a major constraint in order to capture each reflector that must be seen by at least three cameras in the same frame in a sequence of a random motion. The two-dimensional representation is only used to solve the problem of reflector recovery. The choice of two-dimensional representation is to reduce the resolution of a three-dimensional recovery problem to a simple two-dimensional recovery, especially if all the cameras have the same height. With this strategy, the placement of cameras network is not treated as an image processing problem. The use of three variants optimization techniques by Particle Swarm Optimization (Standard Particle Swarm Optimization, Weight Particle Swarm Optimization and Canonical Particle Swarm Optimization), allowed us to solve the problem of cameras network placement in a minimal amount of time. The overall recovery objective has been achieved despite the complexity imposed in the third scenario by the Canonical Particle Swarm Optimization variant.

© 2018 The authors. Published by the Faculty of Sciences & Technology, University of Biskra. This is an open access article under the CC BY license.

1. Introduction

Fu et al. (2014) defined the problem of using the camera network as the way to place the cameras in the appropriate places to maximize the coverage of the camera network under certain constraints. The constraints can be categorized into three main types which are task constraints, camera constraints and scene constraints. Determining the placement of 3D stereo cameras in space, to reduce reconstruction error in the rendered video and to improve the spatial resolution 3D content, have been considered by (Malik and Bajcsy 2008). The reconstruction step is not considered in our work.

In our proposal, the goal is to optimize the placement of a number of cameras (four to ten cameras) in a MoCap system to cover a virtual motion of an object. The purpose of placing a network of cameras is that the network must cover the reflectors placed on the object by three cameras at the same time in each frame of the scene.

The first advantage, for choosing the PSO meta-heuristic in this work, is due to the fact that this technique is a fast technique because of its algorithmic simplicity. The second advantage, which is a few parameters to adjust ((C_{x1}, C_{x2}) position, φ pose, from four to ten cameras, e.g. 12 to 30 parameters) and some constraints to respect (reflector inside the cameras FOV, reflector seen by three cameras in the same frame, critical angle, avoid the obstacle).

The most important contributions of this paper are:

- The optimization of the position and orientation of cameras is based on the use of 2D coordinates of the virtual movement of an object in a MoCap system
- The positions and orientations are optimized according to a main condition, that each reflector is seen by at least three cameras in each frame for all stages of motion.
- The use of three known variants of PSO techniques (SPSO, WPSO and CPSO) to study the performances (recovery of reflectors, minimization of recovery errors and stability of the fitness function) of our proposed MoCap system.
- The convergence performance comparison of three PSO variants. Where, the SPSO and WPSO techniques require several tests to obtain a satisfactory result in terms of internal parameterization. In addition, CPSO does not need adjustment.

The rest of our paper is organized as follows:

Section 2 presents the meta-heuristic PSO method. Section 3 provides a brief presentation of motion capture system. Section 4 describes the study of a two dimensions camera's Field of View. Section 5 gives a detailed explanation of the experimental platform of our application. Section 6 gives the fitness function and constraints. Section 7 summarizes and discusses the results of our camera's optimization system for positions and orientations. Finally, the last section is devoted for the conclusion and perspectives.

[✉] Corresponding author. E-mail address: s.abdesselam@univ-biskra.dz

Nomenclature

$P(P_{x1}, P_{x2})$	Reflector position, cm
$C(C_{x1}, C_{x2})$	Camera position, cm
(d_{max})	maximal depth, cm
(d_{min})	minimal depth, cm
g_{best_j}	Global best solution
$p_{best_{i,j}}$	Personal best solution
$x_{i,j}(k)$	Position
$v_{i,j}(k)$	velocity
C_1, C_2	cognitive and social learning factors
$r_{1,j}, r_{2,j}$	random number in [0,1]
N_c	Cameras number
N	Particles number
D	Problem dimension
N_f	Frames number

Abbreviations

PSO	Particle Swarm Optimization
SPSO	Standard Particle Swarm Optimization
WPSO	Weight Particle Swarm Optimization
CPSO	Canonical Particle Swarm Optimization
2D	two dimensions
3D	three dimensions
MoCap	Motion Capture
FoV	Field of View

Greek letters

f	Fitness function
χ	Constriction coefficient
ω	inertia coefficient
α	angle of view, deg
φ	angle of pose, deg
θ	angle between two adjacent cameras, deg

2. Particle Swarm Optimization

Swarm particle optimization is a metaheuristic technique relies on the stochastic population and an evolutionary computational technique developed by Kennedy and Eberhart (1995) , which mimics the social behavior of flocking bird, fish schooling and herds of animals to find food (Soltani et al. 2013 ; Del Valle et al. 2008).

In the standard PSO algorithm, or SPSO, a "Swarm" consists of N points flying in a D -dimensional search space (i.e., a possible solutions), which are called "Particles". At each iteration k , each particle i is represented by the vector of position $x_i(k) = \{x_{i,1}(k), \dots, x_{i,D}(k)\}$ and its movement is represented by the velocity vector $v_i(k) = \{v_{i,1}(k), \dots, v_{i,D}(k)\}$ of the particles. The relation between these two vectors is given by the position update equation (1):

$$x_{i,j}(k) = x_{i,j}(k-1) + v_{i,j}(k) \quad (1)$$

This means, that the new position of any particle i in iteration k , is calculated by adding a velocity $v_{i,j}(k)$ to the previous position $x_{i,j}(k-1)$. And the vector velocity update is given by equation (2), as follows:

$$v_{i,j}(k) = v_{i,j}(k-1) + C_1 r_{1,j} [p_{best_{i,j}}(k-1) - x_{i,j}(k-1)] + C_2 r_{2,j} [g_{best_j}(k-1) - x_{i,j}(k-1)] \quad (2)$$

For: $i = 1, \dots, N$ and $j = 1, \dots, D$.

This new velocity $v_{i,j}(k)$ depends on the previous particle velocity $v_{i,j}(k-1)$ the distance that the particle is from its

previous personal best solution $p_{best_{i,j}}$ and from the previous global best solution g_{best_j} in the swarm. Where:

$x_{i,j}(k)$ and $v_{i,j}(k)$ are the j^{th} components of the position vector and velocity vector of particle i at iteration k , respectively.

$p_{best_{i,j}}$ is the j^{th} component of a linear attraction towards the best position ever found by the given particle at iteration $k-1$. This component is called "memory" or "self knowledge".

g_{best_j} is the j^{th} component of a linear attraction towards the best position found by any particle at iteration $k-1$. This component is called "social knowledge".

C_1 is the cognitive learning factor that represents the attraction that a particle has toward its own success.

C_2 is the social learning factor that represents the attraction that a particle has toward the success of its neighbors.

$r_{1,j}, r_{2,j}$ are two positive random values with uniform distribution in the range of [0, 1].

In the following, the PSO method algorithm, the role of acceleration constants, inertia weight and constriction coefficient are described in detail.

2.1 PSO algorithm

This algorithm (Algorithm 1) presents in steps, a basic PSO technique for a maximization problem.

Algorithm 1 Particle Swarm Optimization**Start procedure**

Step 1: Initialize a swarm of particles with random positions and velocities on D -dimensions search space

Step 2: Repeat from step (3) to (7)

Step 3: Evaluate the fitness function $f(x_{ij})$ in D variables, for each particle

Step 4: Compare particle's fitness $f(x_{ij})$ with its $p_{best_{i,j}}$.

if $f(x_{ij})$ is better than $f(p_{best_{i,j}})$ then

$$p_{best_{i,j}} \leftarrow x_{ij}; f(p_{best_{i,j}}) \leftarrow f(x_{ij});$$

End

Step 5: Identify the particle that has the best fitness value

if $f(p_{best_{i,j}})$ is better than $f(g_{best_j})$ then

$$g_{best_j} \leftarrow p_{best_{i,j}}; f(g_{best_j}) \leftarrow f(p_{best_{i,j}});$$

End

Step 6: Update the positions and velocities of all particles

Step 7: Until stopping criteria

End procedure

2.2 Acceleration constants

The both cognitive learning factor (C_1) and social learning factor (C_2) determine the balance between the influence of the individual's knowledge and that of the group, respectively. A small variation values limit the movement of particles, while large variation may cause the divergence of the particles. A study on the behavior of the particle influenced by the variation of the coefficients C_1 and C_2 is given by Abdesselam et al. (2016). In general, the maximum value of this constant should be $C = C_1 + C_2 = 4$. A good starting point has been proposed by Kennedy et al. (2001), to be $C_1 = C_2 = 2$.

Some studies indicate that when the maximum velocity and learning factors are properly defined, the particles may still diverge and this phenomenon is called the "explosion" of the swarm. To control this explosion, two methods are proposed in the literature: inertia weight in (Shi and Eberhart 1998), and constriction coefficient in (Clerc and Kennedy 2002).

2.3 Acceleration constants

In 1998, Shi and Eberhart (1998) proposed a modification of the PSO algorithm, aimed at improving its convergence by adding a new parameter ω which will only multiply the velocity at the previous iteration called WPSO for "Inertia Weight". With the implementation of this factor (3), the velocity update equation becomes:

$$v_{i,j}(k) = \omega v_{i,j}(k-1) + C_1 r_{1,j} [p_{best_{i,j}}(k-1) - x_{i,j}(k-1)] + C_2 r_{2,j} [g_{best_j}(k-1) - x_{i,j}(k-1)] \quad (3)$$

The inertia weight can be implemented as a fixed value or can help balance between exploitation and exploration mode through dynamic change. The researchers Shi and Eberhart (1999) have suggested that a linearly decreasing inertia weight can improve the convergence of the PSO algorithm.

2.4 Constriction coefficient

Another variant of PSO technique was developed by Clerc (1999) is called: CPSO for "Canonical". This approach uses the constriction coefficient χ . The use of this coefficient accelerates the particle convergence and this method follows the update (4) equation:

$$v_{i,j}(k) = \chi \cdot \left\{ v_{i,j}(k-1) + C_1 r_{1,j} [p_{best_{i,j}}(k-1) - x_{i,j}(k-1)] + C_2 r_{2,j} [g_{best_j}(k-1) - x_{i,j}(k-1)] \right\} \quad (4)$$

$$\text{Where } \chi = \frac{2}{|2 - C - \sqrt{C^2 - 4C}|}$$

$$\text{And } C = \begin{cases} C_1 + C_2 & \text{if } C_1 + C_2 > 4.0 \\ 0 & \text{if } C_1 + C_2 \leq 4.0 \end{cases}$$

The use of the constriction coefficient will dampen the amplitude of the particle's oscillation and balances the need for local and global search depending on what social conditions are in place.

3. Motion Capture System

The MoCap system, is used for capturing body movements from real life into a computer (motion data), most often in 3D coordinates, using humans as actors as in (McGovern 2009). This system is used in scientific communities, medical (Rehabilitation assessment of injured patients (Li et al. 2012)), engineering, animation industry, sports science, bio-mechanics, monitoring and security (Tirakoat 2011).

The popular system used to capture the movement is the Optical one, called Optical MoCap System as given in (Tirakoat 2011). In this system, the actor is covered with reflectors which are placed in their articulations and tracked by high resolution cameras strategically positioned during the actor's movement (figure 1).



Fig. 1. Optical motions capture system with passive markers.

For each reflector, each camera generates the 2D coordinates, obtained by segmentation step. The data captured by all of the cameras to compute the 3D coordinates of the reflectors are analyzed by appropriate software. A very high sampling rate does not limit the number of reflectors and the freedom offered to the actor’s movement. On the other hand, it suffers from occluded data that is unrecoverable and causes «tracking confusion” if the number of reflectors increases.

Before tracking the object, it is necessary to measure the location and orientation of each camera, which is expensive in time. During monitoring, the intrinsic and extrinsic camera parameters are used to estimate (X,Y,Z) the positional data of the tracking performer's area (Akazawa et al. 2002).

3.1 Criteria for placement of the camera

The camera placement problem can be defined as how to place the cameras in appropriate places to maximize the coverage of reflectors moving in the workspace under certain constraints. These constraints can be divided into three main types: The task constraint includes a complete coverage of the reflectors, the constraint of the camera intrinsic parameters (focus length, Charge-Coupled Diode (CCD) size, etc.) and the scene constraint includes the area (2D, with or without obstacles, etc.). Under these constraints, a basic particle swarm optimization algorithm is proposed to solve the problem of placement of a camera's network.

3.2 Camera field of view modeling

The researchers Morsly et al. (2010) describe the representation of camera’s Field of View (FoV) through its horizontal angle α in 2D plane case (or α_h and α_v in 3D case) and its origin camera’s position. The computation of the projected camera FoV on the 2D plane based on the camera’s intrinsic parameters is described in (Amiri and Rohani 2014).

The field of view of each camera, is inspired by Zhao and Senching (2009), using an isosceles triangle as shown in figure 2. In this section, a network of N_C cameras ($C^i, i = 1, \dots, N_C$) is deployed in a rectangular workspace. Each camera C^i is deployed in its position (C_{x1}, C_{x2}), and posed with a horizontal angle φ with respect to the bisection of the viewing angle α , which defines the aperture of the camera, and finally the maximum depth of the camera d_{max} or the working distance. The area covered by each camera defines a triangle as shown in fig. 2.

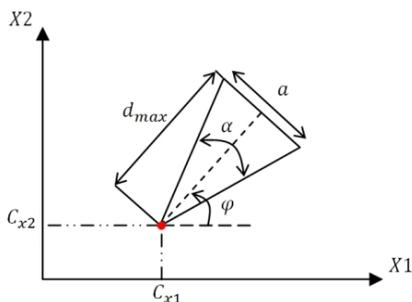


Fig. 2. Field of view parameters in 2D plane.

Thus, a point P positioned at (P_{x1}, P_{x2}) (figure 3) can be visualized by any camera C_i , if the constraints (5) and (6) are satisfied (Morsly et al. 2010):

$$fct_1 \leq d_{max} \tag{5}$$

$$\left(\frac{-a}{2.d_{max}}\right) fct_1 \leq fct_2 \leq \left(\frac{+a}{2.d_{max}}\right) fct_1 \tag{6}$$

Where: $fct_1 = \cos \varphi . (P_{x1} - C_{x1}^i) + \sin \varphi . (P_{x2} - C_{x2}^i)$

$$fct_2 = \cos \varphi . (P_{x2} - C_{x2}^i) - \sin \varphi . (P_{x1} - C_{x1}^i)$$

Two regions of FoV called “Dead Zones” are defined by Amiri and Rohani (2014) and represented by two planes (see figure 3). The plane N , which is closer to the camera, corresponds to the minimal depth (d_{min}) and the plane F , which is farther from the camera, corresponds to the maximal depth (d_{max}). The area which lies between the two planes is the visible region.

3.3 Critical angle constraint

Malik and Bajcsy (2008) define the critical angle constraint (equation (7)), that ensures the angle θ between two adjacent cameras must not exceed $\theta_{i,i+1} = \theta_{min}$. The same point $P(P_{x1}, P_{x2})$ is seen by both cameras (figure 4).

The critical angle is given by the following equation:

$$\theta_{i,i+1} = \arccos \left[\frac{(C^i - P) \cdot (C^{i+1} - P)}{\|C^i - P\| \cdot \|C^{i+1} - P\|} \right] \tag{7}$$

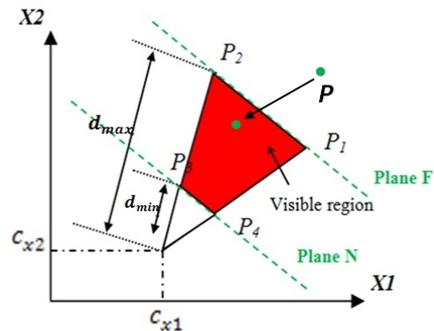


Fig. 3. Trapezoidal visible region (Reflector P will be viewed by camera C if it is inside the red trapezoidal region).

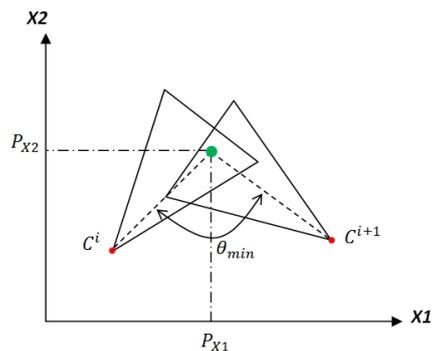


Fig. 4. Critical angle constraint θ_{min} .

With C^i, C^{i+1} : Coordinates of two adjacent cameras.

$P = (P_{x1}, P_{x2})^T$: Coordinates of the point, as seen by both cameras.

3.4 Obstacles consideration

In the real case, it is necessary to take into account the occlusions (due to obstacles, figure 5) which prevent the capture of the reflector by the camera.

In our application, the obstacles are external occlusions (presence of an opaque object between the camera and the reflector) and they are simulated by black segments (S^1, S^2), like walls (figure 5).

The decision that an element is hidden from a camera is that there will be an intersection between the line defined by the ends of the segment (S^1, S^2) and that defined by the coordinates of the point $P(P_{x1}, P_{x2})$ and the camera $C(C_{x1}, C_{x2})$.

4. Experimental Platform

In this platform, some steps are considered such as the modeling of the room and the cube, as well as proposed scenarios:

4.1 Room modeling

A room is rectangular in size (18m×12m), shown in figure 6. The room is equipped with a ten 3D infrared cameras, whose supports are fixed on the walls or a removable tripod. The room is simulated as follows:

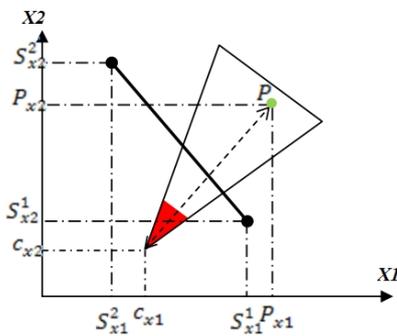


Fig. 5. Obstacle consideration constraint (P is not seen by the camera due to the presence of obstacle segment).

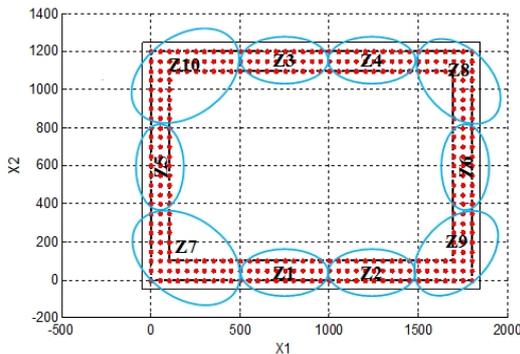


Fig.6. Ten location area organization for cameras (only one camera per zone).

The cameras are placed at the edges of the room where the red dots represent possible camera placements region. In this case, the room is divided into ten areas (one camera for each area), which represent each placement interval of the camera. This platform is designed based on a MoCap system experiment room available at the laboratory LAMIH (Laboratory of Automation, Mechanics and Industrial computing and Human) of the University of Valenciennes in France.

4.2 Cube modeling

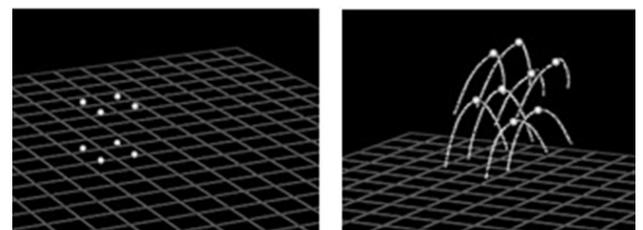
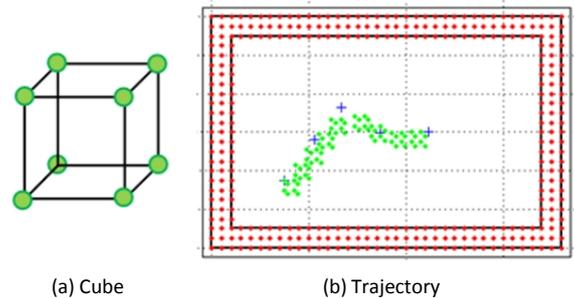
A transparent cube (of dimension 50cm×50cm, figure 7(a)) moves randomly in the scene (figure 7(b)). At the extremities of this cube, are placed eight small reflectors (four in the top and four in the bottom). In the 3D case, the images obtained by the cameras are binary (eight white dots on a black background, figures 7(c) and 7(d), plotted using Mokka free software). These two last figures represent a 3D reconstruction of a cube movement supplied as a three dimensional file by LAMIH laboratory.

4.3 Proposed scenarios

The three different scenarios for the movement of the cube inside the room with the presence of obstacles are proposed to test the robustness of the algorithm.

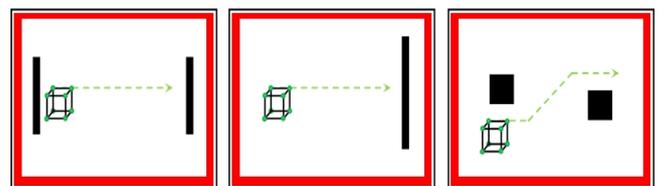
First Scenario: The cube moves in the straight direction between two obstacles (figure 8(a)).

Second scenario: The cube moves in the straight direction in front of a long obstacle (wall) (figure 8(b)).



(c) Reconstruction of the cube shape (d) Movement reconstruction

Fig.7. Random trajectory of the cube.



(a) 1st scenario (b) 2nd scenario (c) 3rd scenario

Fig.8. The three different scenarios.

Third scenario: The cube moves between two pillars (figure 8(c)).

Each scenario is a sequence of video which is divided to a number of frames. In our case, the velocity displacement of the cube increases or decreases the number of frames.

5. Fitness Constraints

5.1 Placement optimization constraints

To formulate the equation of the fitness function (15), a number of variables must be defined. First one, the tensor $t_{ij,k}$ (8) defines which reflector j is inside the FoV of a camera i at the frame k , by:

$$t_{ij,k} = \begin{cases} 1 & \text{if reflector } j \text{ is inside camera's FoV } N^{\circ}i \text{ at frame } k \\ 0 & \text{Otherwise} \end{cases} \quad (8)$$

$t_{ij,k}$ is a $[N_c \cdot N_t \times 3 \times N_f]$ tensor, $j = 1, \dots, N_t$, $i = 1, \dots, N_c$ and $k = 1, \dots, N_f$.

N_c : Number of cameras, $N_c = 1, \dots, 10$.

N_t : Number of reflectors, for our cube $N_t = 8$.

N_f : Number of frames.

The reflector j is judged inside the FoV of a camera i with the presence of obstacle l in between them, if the condition (9) is satisfied:

$$t_{ij,k} = \begin{cases} 1 & \text{if reflector } j \text{ viewed by camera } N^{\circ}i \\ & \text{at frame } k \text{ with obstacle } Obs \\ 0 & \text{Otherwise} \end{cases} \quad (9)$$

The following tensor (10) is simply the middle column of $v_{ij,k}$:

$$O_{i,j,k} = \begin{cases} 1 & \text{if reflector } j \text{ viewed by camera } N^{\circ}i \text{ at} \\ & \text{frame } k \text{ with obstacle } Obs \\ 0 & \text{Otherwise} \end{cases} \quad (10)$$

Where: $O_{i,j,k}$ is a $[N_t \times N_c \times N_f]$ tensor.

Now, the principle condition in our optimization problem is that each reflector must be viewed by only $C_{threshold}$ cameras by frame, and the previous tensor is summed as (11):

$$Object_{j,k} = \sum_{i=1}^{n_c} O_{i,j,k} \quad (11)$$

Where:

$Object_{j,k}$ is a $[N_t \times 1 \times N_f]$ tensor, which represents the number of views of each reflector in all frames.

Finally, equation (12) ensures that the condition $C_{threshold}$ is respected:

$$Object_Fnl_{j,k} = \begin{cases} 1 & \text{if } Object_{j,k} \geq C_{threshold} \\ 0 & \text{Otherwise} \end{cases} \quad (12)$$

$Object_Fnl_{j,k}$ is a $[N_t \times N_f]$ matrix, and provided that each reflector must be seen at least by $C_{threshold}$ cameras (Condition of reconstruction) throughout the scene.

5.2 Penalty functions

Before calculating the fitness of the proposed solution, it must be verified. This solution is penalized by the $Pnlt1$ function (13) when any camera is located outside the room.

$$Pnlt1 = \begin{cases} 1 & \text{if } X1_{\min} \leq C_{x1}^i \leq X1_{\max} \\ & \text{and if } X2_{\min} \leq C_{x2}^i \leq X2_{\max} \\ 0 & \text{Otherwise} \end{cases} \quad (13)$$

Where:

$X1_{\min}$ and $X1_{\max}$ are the ends of the room abscissa.

$X2_{\min}$ and $X2_{\max}$ are the ordered ends of the room.

The second penalty function (14) is related to the condition of critical angle, given by this equation:

$$Pnlt2 = \begin{cases} +\lambda & \text{if } \theta_{i,i+1} > \theta_{\min} \\ -\lambda & \text{Otherwise} \end{cases} \text{ with: } \lambda = 0.5 \quad (14)$$

The value of λ is chosen experimentally.

5.3 Fitness function

The adopted equation of fitness (15) for our application is completed by adding of (16), as following:

$$Ftnes = (fnt1 + Pnlt2) \cdot Pnlt1 \quad (15)$$

Where:

$$fnt1 = \frac{1}{n_t} \cdot \frac{1}{n_f} \sum_{j=1}^{n_t} \sum_{k=1}^{n_f} Object_Fnl_{j,k} \quad (16)$$

6. Optimization and Results

6.1 Optimization proceeding

The simulation begins with the use of four cameras, the choice of the cube path and the installation of obstacles. In case of complete recovery, the results are displayed; otherwise the number of cameras increases and the simulation restarts while keeping all the same parameters.

6.2 Simulation results

In the following figures (9(a), 9(b) and 9(c)), the final solution is given by the optimal positions and orientations of the cameras network for all proposed scenarios. These results are obtained by increasing and decreasing the values of cognitive factors C_1 and C_2 respectively from maximal value (c_{\max}) to minimal value (c_{\min}) according to the iteration steps (Equations (17) and (18)), using CPSO optimization variant. By analyzing these figure, each reflector is covered by at least $C_{threshold} = 3$ cameras (see also Table 1).

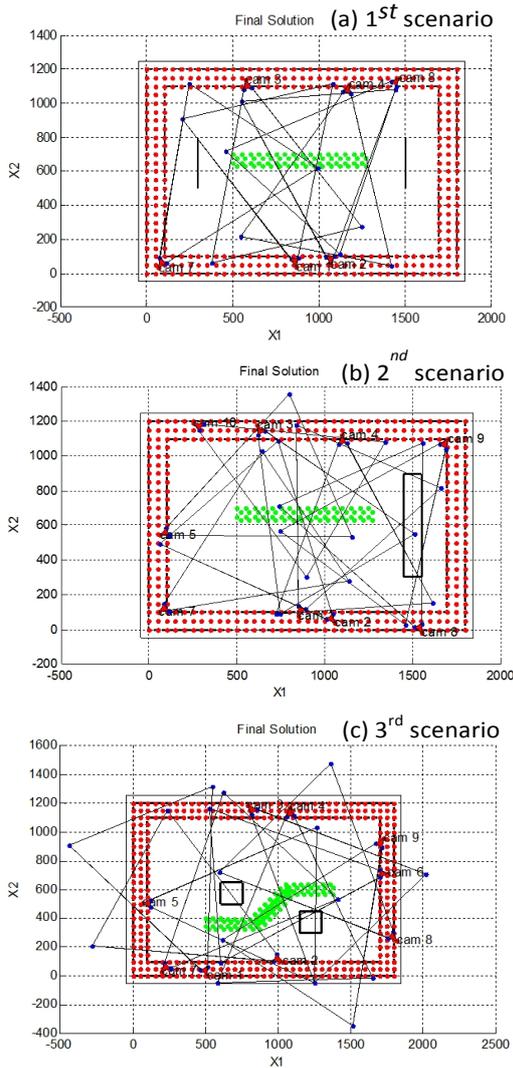


Fig. 9. Optimal placement of a cameras network.

Table 1 PSO Parameters and optimization Results.

	Structured positions		
	1 st Scenario	2 nd Scenario	3 rd Scenario
Particles	30	30	30
Cameras deployed	8	10	9
Cameras used	6	9	9
Camera unused	2	1	0
Frames	8	8	19
$C_{threshold}$	3	3	3
$C1$	1.47	1.02	1.18
$C2$	2.63	3.08	2.92
Recovery [%]	100	100	100

The reconstruction step is easily possible with $C_{threshold} = 3$ (if 3 cameras are used for each frame), and recovery can reach 100%. In the following, only the results obtained for the first scenario are presented for their clarity.

The fitness function given by figure 10, represents the search evaluation of the optimal solution.

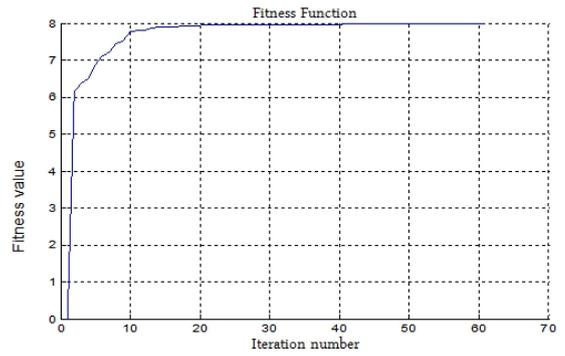


Fig. 10. Evaluation of fitness function.

The figure 11(a), check that the optimal placement is obtained for this situation. In addition, we can follow the evolution of reflector recovery (in percentage), for each iteration of the algorithm. In Figure 11 (b), we show the value of the recovery error and the cancellation of this error at the 24th iteration, but the algorithm continues to progress until it makes sure that the error is stable.

A better representation (as a histogram) of our results is shown in figure 12. The elapsed time for the second scenario is minimal, because there is only one obstacle and only the 5th camera is not used. In the first scenario, the time increases because there are two obstacles and the 5th and 6th cameras are not used. The last scenario, presents a long time of computation according to the number of frames (19 frames) and complexity of obstacles; but in return, all the cameras deployed are used and the number of iterations is acceptable. In all scenarios, if the number of particles $N = 30$ and $C_{threshold} = 3$, the same percentage of coverage is obtained.

These results are obtained by decreasing cognitive factor and increasing the social learning factors with incrimination of iteration. To avoid looking for velocity values, the constriction factor is used with the variation of cognitive and social factors.

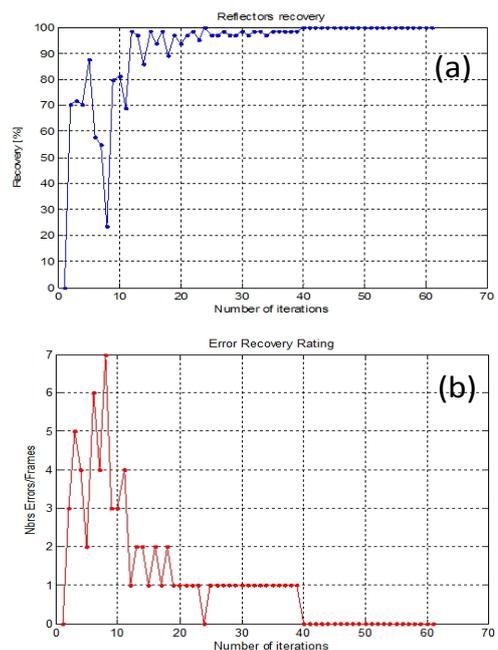


Fig. 11. Evaluation of (a) reflectors recovery and (b) error recovery rating.

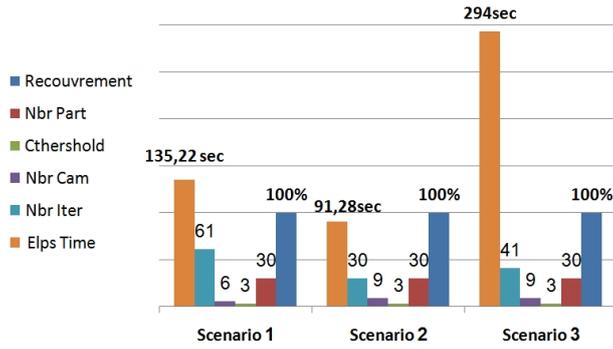


Fig. 12. Histogram of simulation results.

This contribution gives improved results, using the equations (17) and (18):

$$C_1 = \frac{(c_{max} - c_{min}) \cdot t}{t_{max}} + c_{min} \quad (17)$$

$$C_2 = \frac{(c_{min} - c_{max}) \cdot t}{t_{max}} + c_{max} \quad (18)$$

With $c_{min} = 0.6$ and $c_{max} = 3.5$

Thus $C = C_1 + C_2 = 4.1$, respect to (4)

In our application, three different variants of PSO techniques are applied. In order to test the performance of these techniques in a MoCap system, the first scenario is chosen.

In order to avoid the explosion of the system (to leave the search space), in the literature (El Dor, A. 2012), a new parameter V_{max} can be introduced in the SPSO version making it possible to limit the particle speed on each dimension. This method makes it possible to control the divergence of the algorithm and thus to reach an effective compromise between intensification and diversification.

The WPSO version introduced by Shi and Eberhart (1999) balances between local (exploitation) and global (exploration) search. To facilitate global exploration, it is necessary to increase the value of w , while a small value facilitates local exploration.

According to equation (4), the CPSO technique has a fixed χ constriction coefficient. The use of this coefficient makes it possible to better control the divergence of swarm and to get rid of the use of V_{max} (El Dor, A. 2012 ; Clerc and Kennedy 2002).

The curves (figures 13(a), 13(b) and 13(c)) show the development of coverage by application of these variants of PSO techniques.

For SPSO variant (figure 13(a)), the coverage percentage (CP) of reflectors for various number of particles is presented. The results present a great divergence when the number of particles is less than 150 particles. The increase in the number of particles leads to increase a better probability of solution. Despite the use of a large number of particles, the total recovery is not achieved yet ($CP \approx 92\%$), even with the use of all cameras and 300 particles that consume a significant computing time.

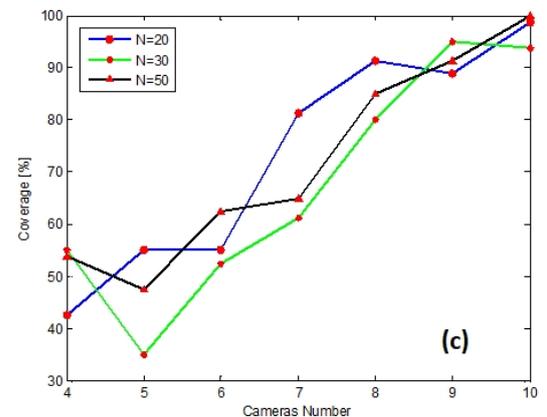
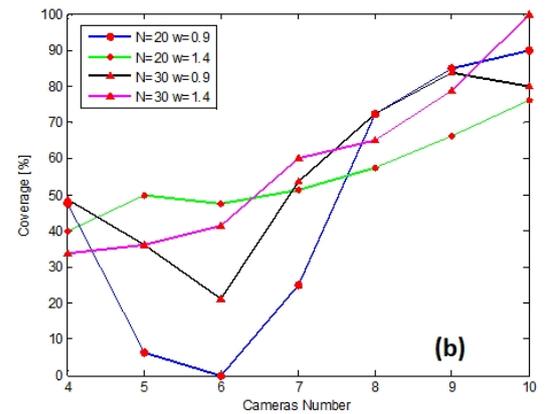
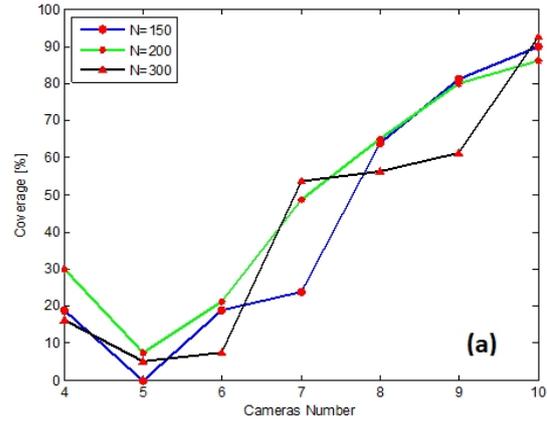


Fig. 13. Reflector coverage percentage evolution in terms of different camera numbers: (a) SPSO, (b) WPSO and (c) CPSO techniques.

With the use of the second variant (WPSO), there is an improvement of the coverage but using: $N = 30$ (number of particles), and $w = 1.4$. In this case, all the reflectors are covered ($CP = 100\%$), see figure 13(b). The third variant (CPSO), presents an upturn in our results ($CP = 100\%$, figure 13(c)) with $N = 50$ particles but using the ten deployed cameras. Also, in this case the total recovery is closer with $N = 20$ particles.

For all the different results, there is a small divergence in the case of using five and six cameras. It is explained by an exhaustive search for a solution in an impossible place (big occlusion seen by the two cameras 5 and 6) for the PSO algorithm. In such circumstances, it is advisable to use a previous step to avoid these dead zones, where finding a solution is almost impossible.

7. Conclusion and Perspectives

In this work, three variants of PSO techniques (SPSO, WPSO and CPSO) are applied in a MoCap system that solves the problem of placing a network of cameras in a simple 2D environment. In addition, there is an improvement in the performance of the third variant technique (CPSO) with the implementation of changes in cognitive learning factors to update the new velocity factor.

There is some coverage optimization work using the PSO technique in literature, but it involves covering a surface for monitoring tasks as in (Morsly et al. 2010) or for cover a workspace like in (Malik and Bajcsy 2008). On the other hand, our work consists of covering particular points under the condition of a possible reconstruction of motion in 3D. In a future work, the use of a 3D field of view is proposed to increase the dimension D of the optimization problem (position 3D (X,Y,Z), two poses (φ_v and φ_h), two depths (d_{min} and d_{max}), two aperture angles (α_v and α_h) from four to ten cameras).

References

- Abdesslam, S., A. Betka, A. Toumi, Z. E. Baair (2016) Application of Stand-PSO Technique for Optimization Cameras' 2D Dispositions in a MoCap system. *Journal of Applied Computer Science & Mathematics*, 10(21) : 9-16.
- Akazawa, Y., Y. Okada, K. Nijima (2002). Real-time video based motion capture system based on color and edge distributions. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on* 2: 333-336.
- Amiri, M. R. S., S. Rohani (2014) Automated Camera Placement using Hybrid Particle Swarm Optimization. Master thesis, School of Computing Blekinge Institute of Technology, Karlskrona, Sweden.
- Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* 3: 1951-1957.
- Clerc, M., J. Kennedy (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1): 58-73.
- Del Valle, Y., G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, R. G. Harley (2008). Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on evolutionary computation* 12(2) : 171-195.
- El Dor, A. (2012) Perfectionnement des algorithmes d'optimisation par essaim particulaire: applications en segmentation d'images et en électronique (Doctoral dissertation, Université Paris-Est).
- Fu, Y. G., J. Zhou, L. Deng (2014) Surveillance of a 2D plane area with 3D deployed cameras. *Sensors* 14(2): 1988-2011.
- Kennedy, J., R. C. Eberhart (1995) Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, Perth, Western Australia, 5: 1942-1948.
- Kennedy, J., Eberhart, R. C., Shi, Y. (2001) *Swarm Intelligence*, Morgan Kaufmann Publishers. Inc., San Francisco, CA.
- Li, J., J. Yang, Z. Xu, J. Peng (2012) Computer-assisted hand rehabilitation assessment using an optical motion capture system. In *Image Analysis and Signal Processing (IASP), 2012 International Conference : 1-5*.
- Malik, R., P. Bajcsy (2008) Automated placement of multiple stereo cameras. In *The 8th workshop on omnidirectional vision, camera networks and non-classical cameras-OMNIVIS : 1-14*.
- McGovern, N. Y. (2009) Technology responsiveness for digital preservation: a model (Doctoral dissertation, UCL (University College London)).
- Morsly, Y., M. S. Djouadi, N. Aouf (2010). On the best interceptor placement for an optimally deployed visual sensor network. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on : 43-51*.
- Shi, Y., R. Eberhart (1998) A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on : 69-73*.
- Shi, Y., R. C. Eberhart (1999) Empirical study of particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on (3) : 1945-1950*.
- Soltani, I., M. Sarvi, F. Salahian (2013). Various Types of Particle Swarm Optimization-based Methods for Harmonic Reduction of Cascade Multilevel Inverters for renewable energy sources. *International Journal of Innovation and Applied Studies* 2(4) : 671-681.
- Tirakoat, S. (2011) Optimized motion capture system for full body human motion capturing case study of educational institution and small animation production. In *Digital Media and Digital Content Management (DMDCM), 2011 Workshop : 117-120*.
- Zhao, J., S. C. Sen-ching (2009). Optimal visual sensor planning. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on : 165-168*.