

# SWARM OPTIMIZATION FOR TRIDIMENSIONAL POINT CLOUD RECONSTRUCTION USING SUPERSHAPES

Remmach Hassnae

Phd Student, Department of Computer Science, Cadi Ayyad University,  
Bd, Prince Moulay Abdellah, Marrakesh, 46000, Morocco  
remmach.hassnae@gmail.com

Sadgal Mohamed

Professor, Department of Computer Science, Cadi Ayyad University,  
Bd, Prince Moulay Abdellah, Marrakesh, 46000, Morocco  
sadgal@uca.ma

El Fazziki Aziz

Professor, Department of Computer Science, Cadi Ayyad University,  
Bd, Prince Moulay Abdellah, Marrakesh, 46000, Morocco  
elfazziki@uca.ma

**Abstract** - The reconstruction of 3D objects based on point clouds is a difficult and poorly posed problem. This problem is fundamental for many applications such as recognition and identification of objects, robotic navigation, medical diagnostics, etc. In this paper, we propose to reconstruct a 3D object from its 3D point cloud using swarm intelligence algorithms. In our approach, the 3D object is modeled by recent primitives called supershapes; representing the extension of superquadrics with a more prominent limit of demonstrating complex objects. The fundamental idea of our work is to consider the issue of three-dimensional reconstruction as an optimization problem, and then model a 3D object using new primitives. Thus, our approach benefits from the efficiency and robustness of metaheuristics, and the diversity of supershapes. We adjusted and adapted the following swarm intelligence algorithms: artificial bee algorithm, chicken swarm algorithm; for the purpose of tridimensional reconstruction. The employed error function is based in its definition on the Euclidean radial distance. The choice of this function is very important since it will subsequently assume the role of dissimilarities indicator between the initial and reconstructed object. Our approach has demonstrated to be extremely compelling, and has prevailing with regards to giving excellent outcomes, in comparison with other existing methodologies in the literature.

**Keywords:** 3D reconstruction; point cloud; supershape; swarm optimization; metaheuristics; artificial bee algorithm; chicken swarm optimization.

## 1. Introduction

Nowadays, information in its various types; text, pictures, sound, 3D objects, and others; serve as an undeniable resource for a multitude of computer applications in particular in Artificial Intelligence. Thus, the processing of natural 3D objects occupies an important place in computer vision and covers several technological problems related to the image acquisition, the reconstruction, the recognition, etc.

The primary trouble in 3D objects processing shows up at the acquisition level, where you must have as much information as possible. Indeed, there are data acquisition tools on 3D objects that provide incomplete or biased data and require considerable reconstruction work.

A wide array of 3D acquisition approaches and techniques are available to acquire 3D models of real objects [1]. Given the large choices of acquisition systems available today, several types of data are device depending. To handle the various sets of provided data, computer vision uses notable 2D/3D imagery techniques for reconstruction, modeling, and exploitation. The three-dimensional reconstruction that intrigues us in this work is considered as one of the significant components in the process of 3D objects exploitation: representation, segmentation, classification, recognition, and so forth. The reconstruction operation can be complex depending on the nature of the digitized data which can be in a point cloud form, depth maps (stereoscopy), etc. Three-dimensional reconstruction can provide a volume representation (set of Voxels), or a 3D surface constituting the object envelope. In this paper, we are interested in the processing of a 3D point cloud acquired by a device like 3D scanner to have a surface representation.

Even though 3D reconstruction is a very active research subject in computer vision, and a large variety of approaches offering reconstruction from a 3D point cloud, the precision and speed constraints remain the big challenge of any method. The problem becomes even more difficult to solve when it comes to deal with complicated and large data.

The proposed approach for the surface representation consists of using, on one side, new parameterized surface forms called supershapes [2]; recently introduced in the literature to model a 3D object. These shapes are kind of superquadrics [3] generalization, defined by six parameters. Indeed, this number of parameters will make it conceivable to extend the selection of shapes that can represent a 3D object, and thus increase the chances to find an adequate approximate shape. On the other side, our approach proposes to perceive the reconstruction issue as an optimization problem and to move away from traditional methods. There are very few works in the literature touching on three-dimensional reconstruction under this vision and using supershapes given their recent appearance.

Different families of optimization methods exist in the literature. The metaheuristics [4] are reputed to be quite robust in solving difficult optimization problems and are also known for their ease of implementation and their outcome relevance. Thus, we opted the resolution of the 3D reconstruction problem using metaheuristics. We are particularly interested in population-based metaheuristics methods, and more especially in swarm intelligence algorithms [5]. This family of algorithms is inspired from natural biological phenomena.

The main idea is to transform a point cloud, acquired from the object surface, into the closest surface to its real shape. Concretely, it is a matter of finding the shape that perfectly matches the provided 3D point cloud, by using the optimization methods. This work adjusts and adapts some swarm intelligence algorithms, in order to apply it to three-dimensional reconstruction of 3D real objects using supershapes. To illustrate our point, we present in this paper, an optimization model including several swarm algorithms. Out of that, we can thereafter assess and compare their results in terms of accuracy and calculation time.

This article is structured in four fundamental parts. The following first part presents a state of the art concerned with concepts evoked in the proposed solution, the emphasis is on metaheuristics methods, and on 3D reconstruction methods from a 3D cloud existing in the literature. The subsequent part describes, on one side, the methodology adopted to solve the problem by depicting each swarm algorithm used in our approach. On the other side, it presents the recent primitives, called supershapes, that are used here to model the 3D object. In the same section, a detailed description of the proposed approach is presented. The third part of the article reports the experimental results got during the application of the approach on various supershapes, and then analyzes and interprets them. At the end of this article, we discuss the performance of our approach, and raise different perspectives for our future work.

## **2. State of Art**

### **2.1. Acquisition System**

Each category of acquisition system is different and has its own properties. They have been classified into different families by several authors in the literature. Based on their different studies, we provide here a diagram of 3D acquisition systems classification (see Fig.1):

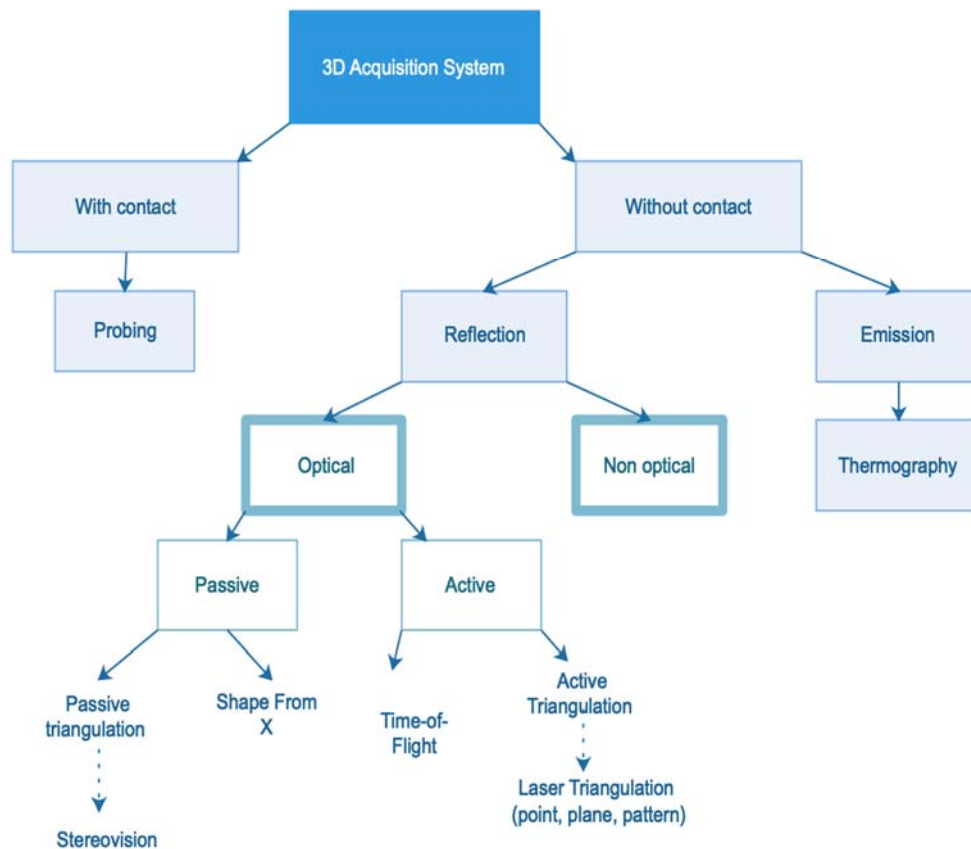


Fig. 1 Classification diagram of 3D acquisition systems.

Some authors [1] [6] have divided 3d acquisition systems into two families: acquisition systems equipped with a sensor with contact (with the object surface), and others equipped with a sensor without contact (with the object surface). Then, we can distinguish two other categories of acquisition techniques [7] [8]. The first category uses special lighting, which the acquisition device emits and subsequently detects its reflection, in order to facilitate the of three-dimensional reconstruction, this is called active techniques. The second category emits no lighting and is based in its 3D reconstruction on detecting reflected ambient light, it is called passive techniques. Their performance strongly depends on the object type and the purpose of use.

## 2.2. 3D Object Representation

The real world is made up of objects with different forms, structures, components, curvatures, and so on. Each 3D object contains a large volume of information that is difficult to handle in processing operations. The researchers attempted to find models simplifying the 3D object representation by minimizing the information loss.

In order to carry out the mission of three-dimensional representation of 3D objects, and to succeed in modeling them correctly, a large choice of methods have been proposed in the literature. The most popular methods for representing a 3D object are polygonal and parametric surfaces.

Polygonal surfaces are designated to model easily simple Euclidean surfaces. Parametric surfaces are approximations of surfaces using curves that generate patches, which will be manipulated later to reconstruct the surface in question. The most common parametric surfaces for modeling a 3D object are quadrics and superquadrics. Quadrics surfaces [9] are defined in 2D space by quadratic equations. They are used as primitives in several graphic's packages, thus contributing to the 3D reconstruction of complex objects. The superquadrics [3] are the generalization and the extension of the quadric surfaces, obtained by adding two parameters controlling the latitudinal curvature and the longitudinal curvature, which are called the form factors. They were designed to assume the function of basic surfaces in the field of computer vision [10] [11]. They are characterized and distinguished by their simple mathematical representation. Recently, Gielis [2] proposed new three-dimensional surfaces based in their composition on superquadrics, these surfaces are called supershapes. These are the surfaces that we will use later in our approach to model the 3D objects.

### 2.3. 3D Reconstruction from Point Cloud

The first basic 3D-reconstruction methods proposing to establish the three-dimensional reconstruction from a point cloud were divided into two families: the implicit reconstruction methods [12] [13] and the parametric reconstruction methods [14] [15]. After few years, other methods were developed attempting to combine the techniques of the two families [16]. Based on [17], we can classify the most common 3D reconstruction methods from point cloud as follows (see Fig. 2):

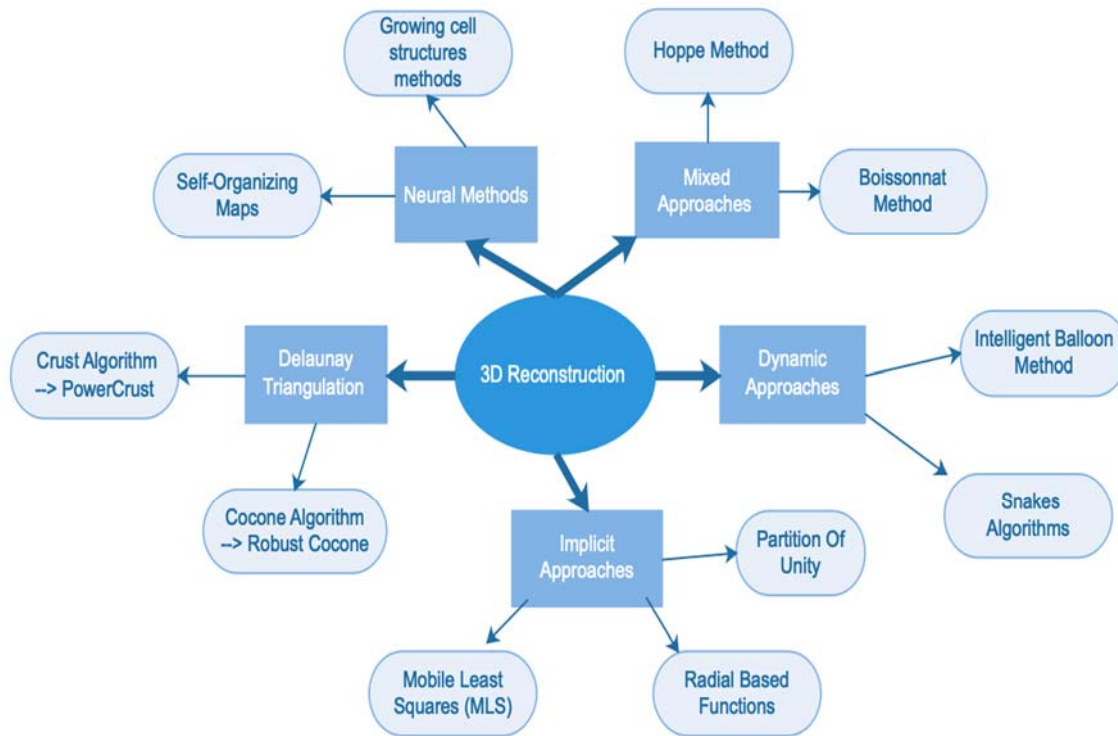


Fig. 2 3D reconstruction methods from point cloud.

Implicit approaches consist in using an implicit representation of surfaces and are based on potential functions, which determine the position of each point in space with respect to the reconstructed surface. Muraki [13] proposes to use implicit surfaces generated by skeletons, which are simple geometric primitives. The idea of this approach consists in placing the skeletons randomly in the point cloud we seek to reconstruct and proceeding with its reconstruction by optimizing the parameters of these skeletons according to the distance separating the implicit surface and the input points. Bittar [18] proposes an improvement of this approach and tries to reduce the computation time so as to be applied to complicated surfaces as in the latest work [19]. Since the 3D object reconstruction from a point cloud employing an implicit approach can be perceived as a constraint approximation or interpolation problem, Turk and O'Brien [20] introduced the radial basis functions which are efficient for solving such an interpolation problem according to Franke's study [21]. This method demonstrated its efficiency and provided good results when it comes to treat a non-voluminous data. Several researchers proposed alternatives to face this major drawback, by introducing additional data processing techniques. For instance, the method of partition of unity based on the principle of "divide and conquer". The application of this concept on 3D reconstruction resides in the division of the point cloud by delimiting it in an enclosing cube. Then, we recursively subdivide this cube enclosing the set of points based on the octree. The interest of this subdivision is summed up in having the option to deal with the issue locally by approximating each cell of the octree, so the final form can be obtained through the fusion of all the local solutions founded. Ohtake et al. [22] propose in their work to set up this fusion using weighted sums; having quadratic functions as weights. While Tobor [23] has opted for the alliance of partition of unity concept and radial-based functions. In this way, these two approaches will make the process of large points clouds possible, and thus remedy the problem of high computation time caused by the large number of points contained in the given data. MLS 'Mobile Least Squares' were introduced for the first time in the literature by Lancaster and Salkauskas [24], these surfaces are without meshes defined by an MLS projection operator. Levin [25] resumed the study of MLS surfaces to demonstrate, in his work, their efficiency in interpolation, derivative approximation and smoothing a set of data points. Several recent research and improvements have been established on these surfaces. First, the MLS surfaces were used in point-based graphics [26] [27], but being defined by traditional projection operator, the computation time was very high. An approach

using a less difficult projection operator has been proposed in [28]. Other works were subsequently directed to evaluate the effectiveness of this operator [29].

Delaunay triangulation is among the primary methods proposed for three-dimensional reconstruction [30]. Many 3D reconstruction algorithms based on the Delaunay triangulation subsequently showed up. Among these algorithms having succeeded in three-dimensionally reconstructing surfaces and having stood out the most from other existing algorithms by providing fairly reliable results, we can cite the Crust [31] and the Cocone [32] algorithms. Both algorithms were introduced into the literature by Amenta with the collaboration of other researchers. Recently improvements of the two algorithms have been proposed: Power Crust [33] and Robust Cocone [34]. The major drawback of methods based on Delaunay triangulation is their vulnerability to noise possibly present in points cloud, as well as the complexity of the calculations when it comes to process a very large point cloud.

There is also the family of mixed methods, we quote for example, the method of Hoppe [35] [36] which uses the computation of iso-values in its process of reconstruction. There is also another method called the Boissonnat method [37] which is based on the Delaunay diagram and the implicit surfaces in its reconstruction process, thus combining two families of reconstruction methods. Although the performance of these mixed methods is acceptable, the problem of computational complexity in processing large data persists. And there is also the family of dynamic approaches characterized in their reconstruction process using meshes, which deform and adapt as well as possible to approach the point cloud. Among the first algorithms of this category of methods, we find algorithms based on active contours (snakes) [38]. Recently, several works have been carried out by Qin, thus developing different algorithms based on dynamic techniques. The Intelligent Ballon method [39] presents better results of 3D reconstruction from a point cloud. These methods are easy to implement and their results are very optimal. However, they encounter great difficulty in maintaining the topology of the model during the deformation process. Besides this major drawback, the computation time consumed is quite high.

Another family recently appeared in 3D reconstruction with promising results, it is about neural methods. These are learning-based methods. There are several neural models in the literature, self-organizing maps were one of the primary neural methods initially used in 2D reconstruction [40], and then developed by several researchers with the aim of solving the 3D reconstruction problem. Kohonen [41] introduced the SOM 'Self Organizing Maps' in the literature. Next, Fritzke [42] defined another competitive learning neural network called Growing Cell Structures, which has been used by other researchers to lead to a fairly robust 3D point cloud reconstruction method [43], in comparison with the results obtained in the work of [36]. It has also been shown in [42] that this method solves the problem of using very large point cloud and overcomes the sensitivity of the algorithm to parasites that may be present in the processed data. However, the constraint of execution time remains present in these methods, because it strongly depends on the number of vertices used in the neural mesh.

#### **2.4. Metaheuristics**

Optimization can be defined as being the operation of searching the solution or an approximate value of the solution of a given problem, in a search set made up of a set of potential solutions. The search for the optimal solution is established by minimizing or maximizing an objective function, called also a fitness or error function.

There are several optimization methods in the literature. They can be classified according to the nature of the issue to be solved [44]. Our concern is a static mono-objective optimization problem. This family has different types of methods including the heuristic methods. These are approximate methods of combinatorial optimization also called discrete, qualified as difficult optimization methods. They consist of finding a value close to the solution of the issue in a polynomial time. There are two types of heuristic methods: deterministic methods [45] and stochastic methods [46]. The difference between these two types lies in the fact that deterministic methods, unlike stochastic methods, are based on perfect knowledge of all the data related to the presented problem. The metaheuristic family is a fairly robust family of heuristic stochastic methods, that can be applied to different kind of problems. Metaheuristics are characterized by their efficiency, their ease of implementation and their correct execution time. They are very effective in solving problems that encounter difficulties in finding the best possible solution of the whole search space, called a global optimum, or the best solutions of the reduced subspaces of the search space, called local optimum, and presents other interesting advantages, we quote random initialization and the low chances of ending up on a local optimum.

Metaheuristics are iterative methods divided into two large families: population-based metaheuristics and single-solution metaheuristics. The difference between these two families lies in the fact that single solution metaheuristics exploit the search space in order to find the optimum, by following the iterative evolution of a single solution. In contrast, population-based metaheuristics manipulate, in their search, a population of solutions to explore the search space, and thus increase the chances of obtaining the best possible approximate solution. Generally, metaheuristics are based in their definition on analogy with physics, with ethology (swarm optimization); for instance, we quote the particle swarm optimization [47] and the ant colony algorithm [48]; or with biology (class of evolutionary algorithms), for example the genetic algorithm [49].

### 3. Methodology

In this article, we propose a solution to the problematic of 3D object reconstruction from their 3D point cloud provided by an acquisition system. The proposed approach is based on an optimization model using population-based metaheuristics.

#### 3.1. General Resolution Architecture

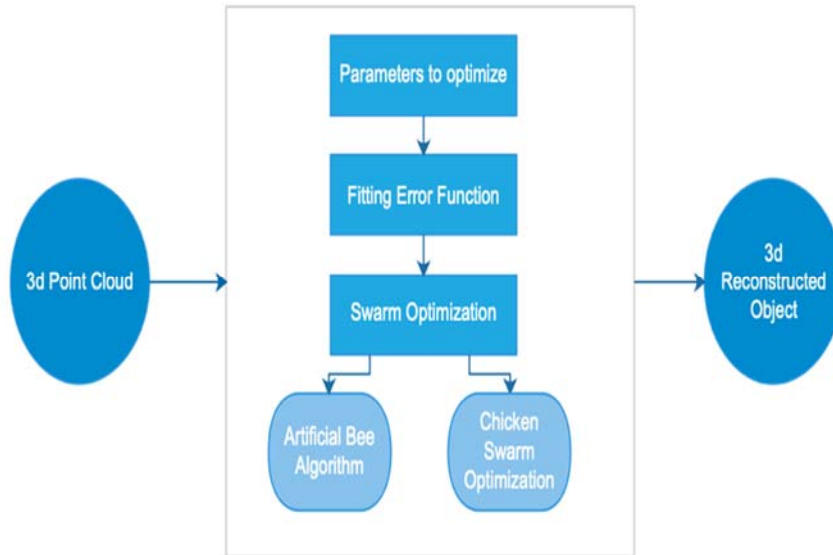


Fig. 3 General architecture.

The approach consists first of fixing the target parameterized surface model, which in our case is the supershapes form. Then, it employs swarm algorithms to determine the best parameters for the generated surface, approaching the initial object represented by the 3D point cloud. This approach aims to optimize the generated 3D object and compare the performance of the used swarm algorithms.

Our approach operates as follows:

- Determine the set of parameters to be optimized (supershapes parameters);
- Define the error function: This must be well defined because it greatly affects the results;
- Apply separately the chosen swarm algorithms: artificial bee algorithm and chicken swarm optimization;
- Analyze and compare the algorithms' efficiency via their results.

#### 3.2. The Supershapes Model

Very recently, Gielis [2] proposed new primitives in the literature as an extension of superquadrics, called supershapes. The primary distinctiveness of these shapes lies in their ability to produce a variety of shapes from a single equation, thanks to their two parameters of symmetry. The second consists of expressing the supershapes in a parametric form and an implicit form.

The parametric representation of supershapes (The expression of point coordinates on the surface) is defined as follows:

$$\begin{pmatrix} x(\theta, \phi) \\ y(\theta, \phi) \\ z(\theta, \phi) \end{pmatrix} = \begin{pmatrix} r_1(\theta)r_2(\phi)\cos\theta\cos\phi \\ r_1(\theta)r_2(\phi)\sin\theta\cos\phi \\ r_2(\phi)\sin\phi \end{pmatrix} \quad (1)$$

$\theta$ : the longitude varies between  $-\pi$  and  $\pi$ .

$\phi$ : the latitude varies between  $-\pi/2$  et  $\pi/2$ .

$r_1$  et  $r_2$ : the rays according to the longitude of the generating superellipses.

Supershapes are the result of the spherical product of two superellipses and can be defined by the following equations:

$$r1(\theta) = \frac{1}{\sqrt[n_1]{\left|\frac{1}{a}\cos\left(\frac{m\theta}{4}\right)\right|^{n_2} + \left|\frac{1}{b}\sin\left(\frac{m\theta}{4}\right)\right|^{n_3}}} \quad (2)$$

$$r2(\phi) = \frac{1}{\sqrt[3]{\left|\frac{1}{a}\cos\left(\frac{M\phi}{4}\right)\right|^{N_2} + \left|\frac{1}{b}\sin\left(\frac{M\phi}{4}\right)\right|^{N_3}}} \quad (3)$$

The longitude and the latitude are defined as follow:

$$\begin{cases} \theta = \theta(x, y) = \arctan\left(\frac{y}{x}\right) \\ \phi = \phi(x, y, z, n_1, n_2, n_3) = \arctan\left(\frac{zr_1(\theta)\sin(\theta)}{y}\right) \\ \quad = \arctan\left(\frac{zr_1(\theta)\cos(\theta)}{x}\right) \end{cases} \quad (4)$$

$a, b \in \mathbb{R}^+$  are the two parameters controlling the dimensions of the generator polygon,  $m, M \in \mathbb{R}_+^*$  are symmetry parameters, the parameters  $n_i, N_i \in \mathbb{R}$  with  $n = 1, 2, 3$  are the shape coefficients of the supershape. In comparison with superquadrics, supershapes have three shape parameters per polynom generator instead of one. In addition to these shape parameters, the supershapes has two symmetry parameters, thus having the advantage of being able to represent a 3D object under different axes of symmetry. Thus, a supershape can be expressed by the following parameters:

$$V = [a, b, m, M, n_1, n_2, n_3, N_1, N_2, N_3] \quad (5)$$

By adjusting this set of parameters, we can get a multitude of simple and complex supershapes. Some examples are shown in the following figure, the values of the corresponding parameters are displayed in Table 1:

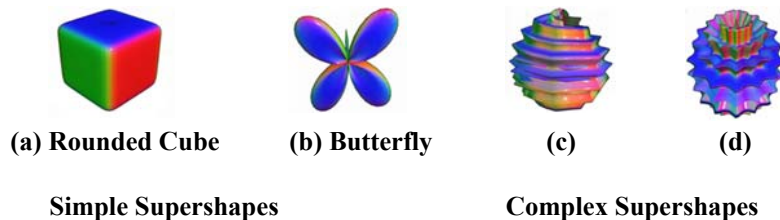


Fig. 4 Samples of Simple and Complex supershapes.

The following table presents all the parameters values described in Eq. (5), it should be noted that the forms concerned are unitary supershapes ( $a=1$  and  $b=1$ ):

Table 1 Parameters of the supershapes presented in fig.4.

	<b>m</b>	<b>n<sub>1</sub></b>	<b>n<sub>2</sub></b>	<b>n<sub>3</sub></b>	<b>M</b>	<b>N<sub>1</sub></b>	<b>N<sub>2</sub></b>	<b>N<sub>3</sub></b>
<b>(a)</b>	4	10	10	10	4	10	10	10
<b>(b)</b>	8.52	33	21.39	-33.40	2.57	0.10	-5.13	1.71
<b>(c)</b>	7.50	1.80	1.50	0.90	16.20	1	1.60	1.10
<b>(d)</b>	12.80	16.80	9.20	10	17.70	7.20	11.10	9.20

### 3.3. The Adopted Optimization Algorithms

The optimization procedure relies on the use of swarm intelligence algorithms. For this, we applied two swarm intelligence algorithms in our approach: Artificial Bee Algorithm and Chicken Swarm Optimization.

#### 3.3.1. Artificial Bee Algorithm (ABA)

The bee is an intelligent and very organized insect in its work. The process used by bees to collect the maximum amount of nectar is based on three types of bees: active forager (employee bee), forager scout (scout) or inactive forager (onlooker). Bees begin the process by randomly traversing the environment to find good quality food sources, that is, sources with a large amount of nectar. Once the scout bee finds the right food source, it becomes an employed or active bee, and begins to collect the food to bring it back to the hive. The latter communicates information about the source by dancing on the dance floor. Thus, the inactive bees observe the active or scout bees, and establish the most judicious choice of the source to exploit. Once the source is exhausted, the bee becomes a scout again in search of new sources of food.

The Artificial Bee Algorithm [50] consists of looking for the location of the food source, which is therefore the solution of the optimization problem. Each food source is occupied by a single employee bee, so the number of possible solutions in the population comes down to the number of active or inactive foragers present in the population.



First, the algorithm starts with an initialization of the population with SN randomly distributed solutions. SN represents population size (number of food sources). Each solution is represented as a vector of solutions  $x_i$  ( $i = 1, 2, \dots, SN$ ). The values of each vector are initialized by the scouts, and must be optimized by evaluating the fitness function of the population as shown in Fig. 5.

```

Begin
Initialization of the population of bees  $x_i$ 
Fitness function evaluation
Memorize the best solution  $x_{best}$ 
While ( $Iteration < Iteration_{max}$ )
    For each employed bee
        Generate new solutions  $v_{ij}$ 
        Evaluate fitness function for each agent
        Selection of the best solution
        Calculate the probability  $P_i$  for each solution
    End For
    For each onlooker bee
        Select a solution depending on  $P_i$  value
        Generate new solutions  $v_{ij}$ 
        Evaluate fitness function for each agent
        Selection of the best solution
        If abandoned solution is identified
            Replace it with new generated solution
        End If
    End For
End While
Memorize the best global solution  $x_{Gbest}$ 
End

```

Fig. 5 Pseudo Code of Artificial Bee Algorithm.

After initialization, the population of solutions is subjected to search cycles  $Iteration = 1, 2, \dots, Iteration_{max}$ . Each cycle involves searching for a food source that contains as much nectar as possible, and then the information is shared. An employee bee or inactive bee keeps in its memory the position of the food source (potential solution), to find a new food source (new solution) through the information communicated in the dance floor by evaluating the quantity of nectar, which represents a value of the fitness function. The inactive bee establishes its choice according to the probability value calculated with the following formula:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (6)$$

Where  $fit_i$  is the fitness value of solution  $x_i$ , which is proportional to the nectar amount from the food source at position  $i$ .

To produce a new food source denoted by  $v_i$  containing more nectar from the old source  $x_i$ , we use the following formula:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (7)$$

Where  $k \in \{1, 2, \dots, BN\}$  et  $j \in \{1, 2, \dots, SN\}$  are randomly chosen indices with BN representing the number of employee bees (which is equal to SN the number of food sources). The value of  $k$  must be different from the value of  $i$ .  $\phi_{ij}$  is a random number in the range of  $[-1, 1]$ , it controls the production of a food source in the vicinity of  $x_{ij}$ .

The abandoned food sources will be replaced by the new food sources generated through Eq. (7).



Each new generated food source  $v_{ij}$  is compared to the old food source  $x_{ij}$ . If the new source contains an amount of nectar greater than or equal to that of the old source, it is replaced by the new source. Otherwise, the old source is kept in the bee's memory.

### 3.3.2. Chicken Swarm Optimization (CHSO)

Chicken swarm optimization [51] is a new algorithm based on the behavior of a chicken swarm during its foraging quest, which appeared very recently in 2014. A swarm of chicken is divided into several groups. Each group is composed of a single rooster, several hens and several chicks. The members of the groups are characterized by their hierarchical order. Each type of member has its own hierarchical behaviors. For roosters, the group leader is referred to the dominant rooster. The latter reserves the priority access to the food source and can also invite his companions to join him. He takes care of chasing any other member not belonging to his group, in case the latter tries to invade his territory. Regarding the hens, they also benefit from the right of priority access to the food source, and they are kept near the dominant rooster. Hens dominate the weaker limbs and protect their food source. Concerning the chicks, they usually stay with their mother hens to ensure their survival and find a food source. Chickens communicate with each other using different sounds to circulate all kinds of information (danger, food source, mating, nesting).

Following the distribution of the chicken swarm's groups, the following conclusion can be drawn: the roosters will have the best fitness value, the bad value will be assigned to the chicks, and the rest of the group will be designated as hens. The following pseudo code describes the different stages of the chicken swarm algorithm:

```

Begin
  Initialization of the population of N chickens
  Evaluate the fitness function of the N chickens
  Update the frequency of the chicken swarm G
  While ( $Iteration < Iteration_{max}$ )
    If ( $Iteration \% G == 1$ )
      Rank the chicken population according to their fitness value
      Divide the swarm into three different groups, and the relationship
      between the chicks and its mother
    End If
    For  $i = 1 : N$ 
      If ( $i == rooster$ )
        Update the position solution following the Eq. (9)
      End If
      If ( $i == hen$ )
        Update the position solution following the Eq. (11)
      End If
      If ( $i == chick$ )
        Update the position solution following the Eq. (12)
      End If
    End For
    Evaluate the solutions and Find the best solution  $x_{best}$ 
  End While
  Memorize the best global solution  $x_{Gbest}$ 
End

```

Fig. 6 Pseudo Code of Chicken Swarm Optimization Algorithm.

In a search space  $D$ , we designate  $N$  virtual chickens in search of a food source. They are depicted by their positions as a function of time  $t$ :

$$x_{i,j}^t \quad (i \in [1, \dots, N], j \in [1, \dots, D]) \quad (8)$$

The best chickens able to spot a food source will have a minimum value of the fitness function. The position of the three members of a group is calculated according to the following formulas:

- For the rooster:

$$x_{i,j}^{t+1} = x_{i,j}^t * (1 + \text{randn}(0, \sigma^2)) \quad (9)$$

With,  $\text{randn}(0, \sigma^2)$  : Gaussian distribution with mean equal to 0 and deviation  $\sigma^2$  :

$$\sigma^2 = \begin{cases} 1 & \text{if } f_i < f_k \\ \frac{(f_k - f_i)}{e^{(|f_i| + \varepsilon)}} & \text{with } k \in [1, N] \text{ and } k \neq i \end{cases} \quad (10)$$

$f$ : value of the fitness function of the corresponding  $x$

$k$ : index of the rooster

$\varepsilon$ : very small constant set to avoid a division error on 0

- For the hen:

$$\begin{cases} x_{i,j}^{t+1} = x_{i,j}^t + S_1 * \text{rand} * (x_{r_1,j}^t - x_{i,j}^t) \\ \quad + S_2 * \text{rand} * (x_{r_2,j}^t - x_{i,j}^t) \\ \text{with } 0 \leq \text{rand} \leq 1 \text{ and } S_1 = e^{\frac{(f_k - f_{r_1})}{(|f_i| + \varepsilon)}} \\ \quad \text{and } S_2 = e^{(f_{r_2} - f_i)} \end{cases} \quad (11)$$

$r_1$ : index of the dominant rooster of the hen's group.

$r_2$ : index of the chicken being able to be a hen or a rooster, it should be noted that  $r_1 \neq r_2$

- For the chick:

$$x_{i,j}^{t+1} = x_{i,j}^t + FL * (x_{m,j}^t - x_{i,j}^t) \quad (12)$$

$x_{m,j}^t$ : determines the position of the  $i$ th chick's mother

$FL$ : parameter indicating that the chick follows its mother during the search for food source, it should be noted that this parameter is chosen randomly  $0 < FL < 2$ .

### 3.4. Approach Description

In this article, we propose to apply the swarm intelligence algorithms to reconstruct a 3D object from its 3D point cloud, using the recent primitives described in the previous part. The 3D reconstruction problem is considered in this paper as being an optimization problem. In order to solve any optimization problem, we must first determine an objective function. Before proceeding with this, it is necessary to delimit the problem to be solved. Indeed, it is a question of finding the shape of the 3D object that we are seeking to reconstruct from its 3D point cloud. Since the object in question is modeled with supershapes, this search then returns to find all the parameters representing the supershape which matches the shape of the given point cloud. The following flowchart describes the process of our approach:

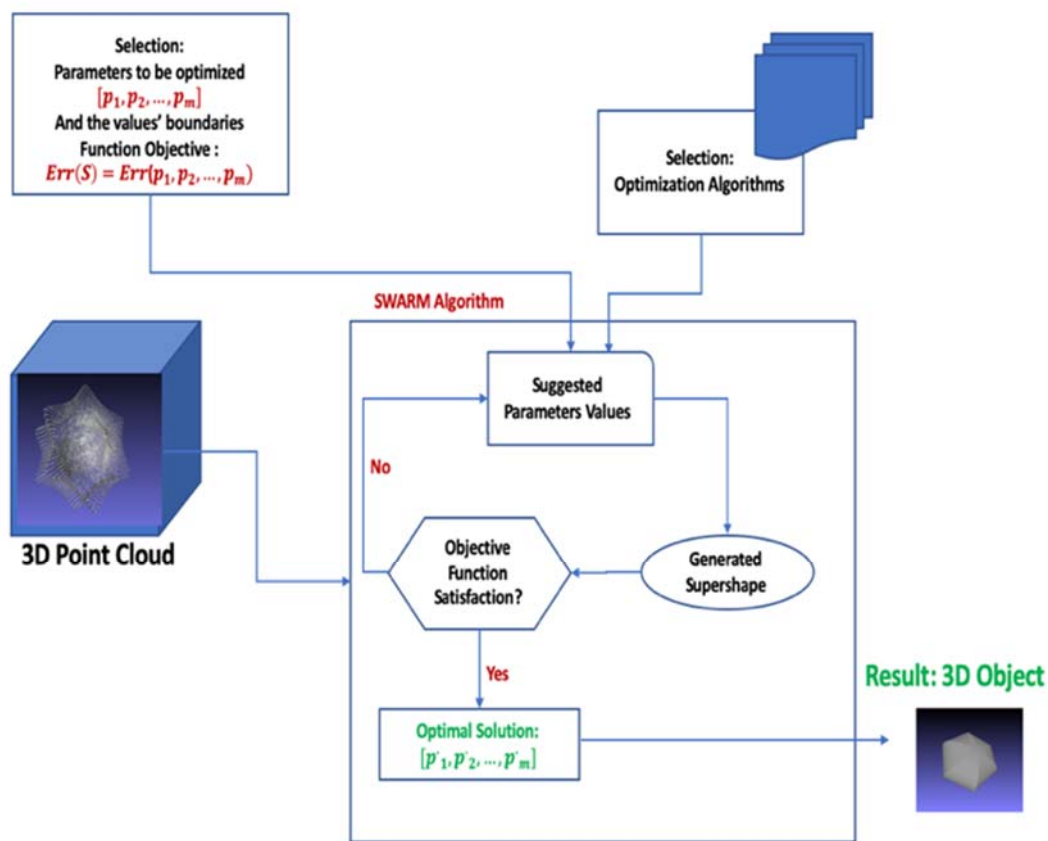


Fig. 7 Flowchart of the proposed approach.

### 3.4.1. Entrance Data

The data to be processed by our approach are 3D points clouds obtained by an acquisition device. It is the most common representation of acquisition systems, because it preserves the shape and scale information of the object as well as its geometric data. A 3D point cloud is quite simply the set of geometric information specific to the object in question, that is to say all of its cartesian coordinates. It is natural that a 3D point cloud of a complex object or an entire scene will be very large. It can include thousands or millions of points. This will be restrictive for our optimization process. The speed of execution constraint will be compromised given the considerable number of points to be processed. To remedy this problem, it is proposed to apply a sampling of the points cloud in order to reduce the number of points. Here is an example of sampling a point cloud from a supershape:

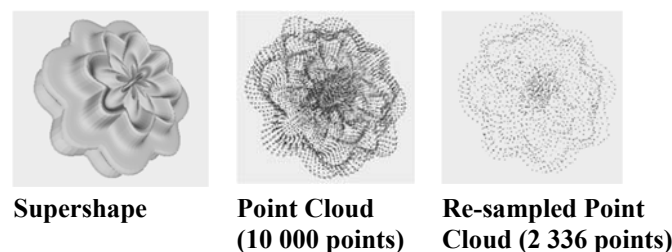


Fig. 8 3D Point Cloud Re-sampling.

### 3.4.2. Definition of Parameters to Optimize

In order to proceed with our optimization process, we must define the parameters we want to optimize. In our case, it is about minimizing a set of parameters in order to coincide with the parameters corresponding to the supershape; which models the 3D object to be reconstructed. A supershape is expressed in terms of a set of ten parameters as shown in Eq. (5) in the previous section. We can introduce classical affine transformations to this equation. And in order to generalize the representation of supershapes, other transformations can be applied. We cite for example the classical deformations defined by Barr [52], known by their small number of required parameters:

- Tapering along the Z axis: this transformation is similar to the affine scaled transformation, except that here the modification will only be applied to two components. By applying tapering to a point  $P(x, y, z)$ , the corresponding new coordinates are expressed as follows:

$$\begin{cases} X = f(z) x \\ Y = f(z) y \\ Z = z \end{cases} \quad \text{with } f(z) \text{ is the tapering function}$$

- Bending along the Z axis: this transformation consists of applying a projection of the two components to be changed on the Bending plane. Then, we redo the projection on the original plane. This operation transforms the coordinates of a point  $P(x, y, z)$  into:

$$\begin{cases} X = -\sin(\theta) * \left(y - \frac{1}{k}\right) + x_0 \\ Y = \cos(\theta) * \left(y - \frac{1}{k}\right) + \frac{1}{k} \\ Z = z \end{cases} \quad \text{with } \theta = (x - x_0) * k \text{ and } k \text{ is the bending rate}$$

- Twisting along the Z axis: this transformation has similarities with the affine rotation transformation. The rotation will be applied on two components according to the global base vector z. By applying twisting to a point  $P(x, y, z)$ , the corresponding new coordinates are expressed as follows:

$$\begin{cases} X = x \cos(\theta) - y \sin(\theta) \\ Y = x \sin(\theta) + y \cos(\theta) \\ Z = z \end{cases} \quad \text{with } \theta = f(z)$$

By applying the classical affine transformations and the Barr deformations on a supershape, its equation is expressed as follows:

$$[a, b, m, n1, n2, n3, M, N1, N2, N3, T_x, T_y, T_z, R_x, R_y, R_z, S_x, S_y, S_z, g_x, g_y, g_z, t_x, t_y, t_z, p_x, p_y, p_z]$$



$T_x, T_y, T_z$  : the translation factors according to the 3 directions  $(x, y, z)$ ;

$R_x, R_y, R_z$  : the rotation factors according to the 3 directions  $(x, y, z)$ ;

$S_x, S_y, S_z$  : the scaling factors according to the 3 directions  $(x, y, z)$ ;

$g_x, g_y, g_z$  : the tapering factors according to the 3 directions  $(x, y, z)$ ;

$t_x, t_y, t_z$  : the twisting factors according to the 3 directions  $(x, y, z)$ ;

$p_x, p_y, p_z$  : the bending factors according to the 3 directions  $(x, y, z)$ .

### 3.4.3. Fitting Error

Several studies have been carried out in order to determine the most suitable error function for superquadrics. Opinions have differed on this subject, some authors assume that the error function based in its definition on pseudo-Euclidean distance is more efficient [53], and others assume that other functions such as interior-exterior functions can be more efficient [54]. On the other hand, these are vulnerable to any noise that may be present in the data to be processed. Therefore, in this paper, we will join the first opinion and propose an error function based on the Euclidean radial distance.

In our case, we seek to find the shape that coincides with the 3d point cloud provided as input data. We decided to approach this form by the recent primitives representing the extension of the superquadrics, called the supershapes. In other words, we must determine the position of the points of the cloud to be processed in relation to the supershape. It is then a matter of calculating the distance separating each point of the cloud from the supershape, thus distinguishing the points belonging to the surface of the supershape. Since supershapes are part of the family of parametric surfaces, the potential function is defined as follows:

$$F(x, y, z) = 1 - \frac{x^2 + y^2 + r_1^2(\theta)z^2}{(r_2^2(\theta) * r_1^2(\theta))} \quad (15)$$

And the best candidate function apt to succeed in computing this distance is this potential function defined by a Euclidean radial distance. This function can also be expressed in the following form according to Fougerolle [55]:

$$F(x, y, z) = 1 - \frac{\|\overline{OP}\|}{\|\overline{OI}\|} \quad (16)$$

The distance between a point  $P(x, y, z)$  and a shape  $S$ , with the point  $O$  as its center (see Fig. 10) is expressed as follows:

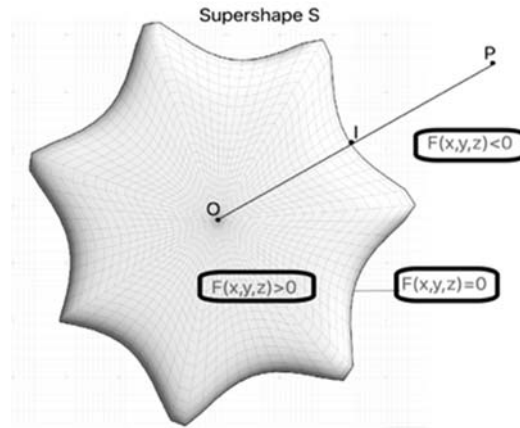


Fig. 9 Illustration of radial distance on a supershape.

By analyzing the sign of the potential function three conclusions can be drawn (see Fig. 10):

- $F(x, y, z) = 0$  : Point P is located exactly on the surface of the supershape.
- $F(x, y, z) < 0$  : Point P is outside the supershape.
- $F(x, y, z) > 0$  : Point P is inside the supershape.

So the error function will be defined as follows:

$$\text{Err}(V) = \sum_{i=1}^n F^2(P_i) \quad (17)$$

$P_i$  represents the point cloud with  $i = 1 \dots n$  and  $n$  is the number of the points present in the point cloud.

The solution to our 3D reconstruction problem from a point cloud will then be summed up in the search for the vector of parameters of the supershape minimizing this error function. The general expression of the error function then becomes:

$$\text{Err}(V) = \text{Err}(p_1, p_2, \dots, p_m) \quad (18)$$

With :  $V = [a, b, m, M, n_1, n_2, n_3, N_1, N_2, N_3]$ , the vector of 10 parameters of the supershape S.

Then in our case, the error function that we are seeking to minimize is expressed as follows:

$$\text{Err}(V) = \text{Err}(a, b, m, M, n_1, n_2, n_3, N_1, N_2, N_3) \quad (19)$$

#### 4. Experimental Result

The proposed approach proceeds by minimizing an error function composed of supershape parameter set, employing the swarm intelligence optimization methods. The choice fell on the following methods: Artificial Bee Algorithm and Chicken Swarm Optimization.

##### 4.1. The Input Data and Parameters

To check that the used algorithms converge to an optimal solution, we applied them on four different point clouds corresponding to four different supershapes:

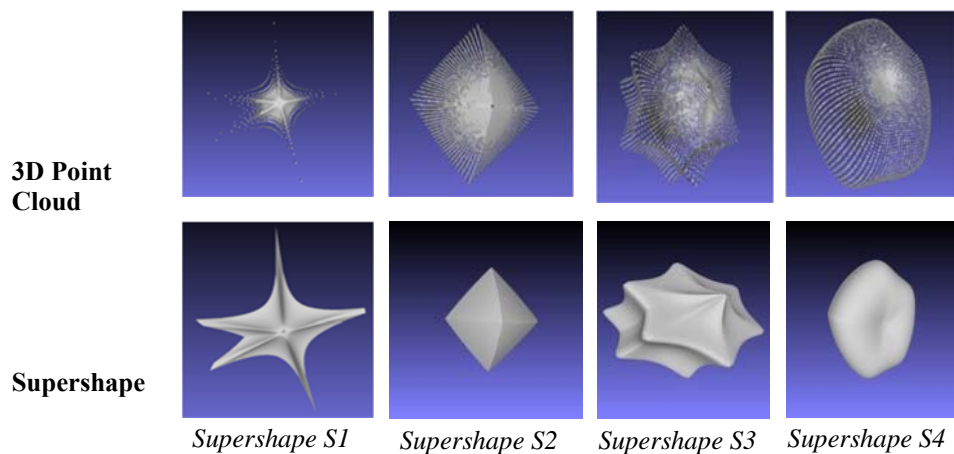


Fig. 10 3D point cloud and its corresponding supershape.

Given the large volume of 3D points clouds supplied as input, it is judicious to reduce the number of points in the cloud by sampling, thus reducing the data to be processed and optimizing the calculation time:

- The S1 point cloud has been reduced from 10,000 to 158 points.
- The S2 point cloud has been reduced from 10,000 to 160 points.
- The S3 point cloud has been reduced from 10,000 to 276 points.
- The S4 point cloud has been reduced from 10,000 to 119 points.

We set the number of iterations at 500. Each algorithm tested adopts different hyper-parameters:

- Parameters used in Artificial Bee Algorithm: we used 80 agents in a search space of dimension 10. These are the standard arguments used in the method.
- Parameters used in Chicken Swarm Algorithm: The number of agents used is 50. The parameters relating to the algorithm are:
  1. The time taken to upgrade the relationship:  $G = 5$
  2. The indicator parameter: the chick would follow its mother to drilling for food:  $FL = 0.5$

#### 4.2. The Results

The following table presents all the obtained results. The values appearing in the table are the minimum values obtained from the defined error function:

Table 2 Fitness error values.

	Artificial Bee Algorithm	Chicken Swarm Algorithm
<b>S1</b>	0.280	0.301
<b>S2</b>	0.075	0.069
<b>S3</b>	0.030	0.076
<b>S4</b>	0.019	0.027

The visual result of the 3D reconstruction of the tested shapes is as follows; accompanied by a curve showing the evolution of the error function value according to the number of iterations for each supershape:

#### Artificial Bee Algorithm

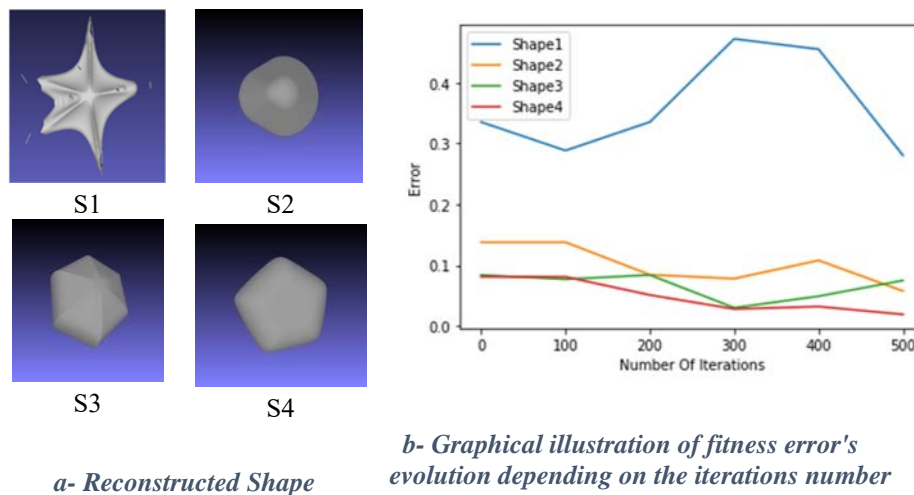


Fig. 11 Results obtained by applying Artificial Bee Algorithm.

The results of 3D point cloud reconstructions acquired by the Artificial Bee Algorithm are sufficient. For all the supershapes, we can notice that the obtained forms affirm the numerical values displayed in table 2 of the fitness function. The best optimal outcome obtained concerns the supershape S4. Its reconstruction is practically identical with the initial shape shown in Fig.11. The supershapes S2 and S3 results are quite satisfactory and correspond to the real shape. The supershape S1 is the shape with the highest error value, therefore its reconstruction shows some imperfections. We then conclude that Artificial Bee Algorithm has provided very good 3D object reconstruction results especially for supershape S4.

## Chicken Swarm Algorithm

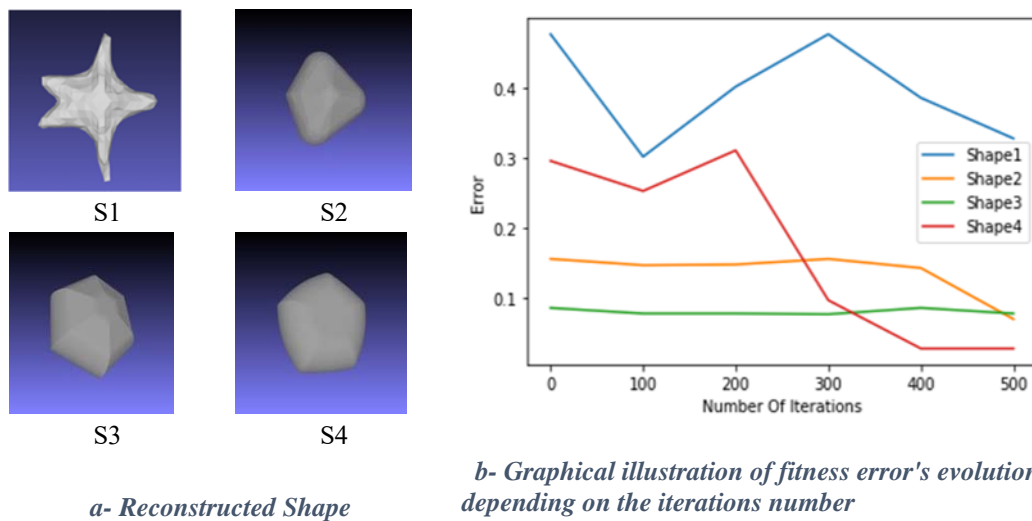


Fig. 12 Results obtained by applying Chicken Swarm Algorithm.

Chicken Swarm Optimization presents suitable results. For all the supershapes, we can observe that visual rendering confirms the numerical value displayed in table 2 of the fitness function. The best optimal result obtained concerns the supershape S4. Its reconstruction perfectly resembles the initial shape shown in Fig.12. The results obtained for the supershapes S2 and S3 are adequate to the real shape. Supershape S1 has the highest error value, therefore its reconstruction has some flaws especially in the sharp parts of the shape, otherwise the reconstruction is acceptable. We then conclude that Chicken Swarm Optimization provided fairly good 3D point cloud reconstruction results, more particularly for supershape S4.

Our approach consists of finding a solution for three-dimensional reconstruction problem of a 3D object from its 3D point cloud. It has been proposed to solve this problem, on the one hand, throughout swarm intelligence algorithms, and on the other hand by modeling 3D objects with recent parametric forms, called supershapes. The choice fell on the following algorithms: Artificial Bee Algorithm (ABA) and CHicken Swarm Algorithm (CHSO).

First, we compare the performance of the algorithms with respect to each chosen form. For this, we have drawn up the plot of the curves' evolution of the fitness function value according to the number of iterations, for each algorithm applied to the four supershapes tested (see Fig. 13).

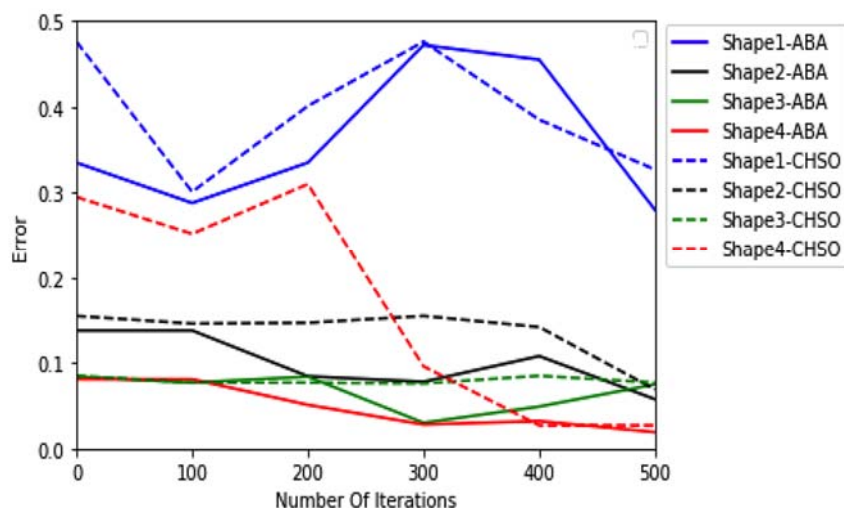


Fig. 13 Comparative plot of error evolution curves.

The choice fell on four supershapes with completely different details. From results, it seems that the supershape S1 is the most difficult form to rebuild. Artificial Bee Algorithm presented the best reconstruction result for this complex shape, and for the shapes S3 and S4. Moreover, the shape S4 can be voted as being the easiest shape to reconstruct, since its fitness values are the smallest in comparison to all the other shapes. As for the shape S2, Chicken Swarm Algorithm displayed the best possible reconstruction. We can then conclude that



Artificial Bee algorithm has shown more efficiency and insight. The processing time is an important factor in the performance of an algorithm. By classifying the tested algorithms in descending order according to the required time, the Artificial Bee Algorithm (ABA) comes first, and CHicken Swarm Optimization (CHSO) comes second. As we can notice, the ABA algorithm presents better results than the CHSO algorithm, but the latter is faster. So, the qualification of their performance remains relative.

By browsing existing works in the literature, we find very little papers dealing with the same problematic and especially employing supershapes.

We compared the performance of the swarm intelligence algorithms tested with two other different types of algorithms: a deterministic Levenberg-Marquardt algorithm [56] and to an evolutionary algorithm in this case the genetic algorithm [57]. As well as with the PSO [58] algorithm belonging to the same family of swarm intelligence algorithms. To have a good visibility, we have drawn up a table (see table 3) displaying the set of results obtained with all the algorithms concerned on the different supershapes used.

Table 3 Comparative Table.

	ABA	CHSO	LM Algorithm	Genetic Algorithm	Particle Swarm Optimization
<b>S1</b>	<b>0.280</b>	0.301	32.5	2.017	-
<b>S2</b>	0.075	<b>0.069</b>	1.35	0.76	0.32
<b>S3</b>	0.030	0.076	0.02	0.0007	-
<b>S4</b>	<b>0.019</b>	0.027	3.54	0.24	0.10

There is a glaring difference in the error values relating to the supershape S1. The algorithms used in our approach present better values and an excellent performance for the reconstruction of the shape S1. Concerning the shapes S2 and S4, we notice a very considerable difference in the values obtained. As for the S3 form, the LM algorithm and the genetic algorithm both show more optimal values compared to our values. One in four shapes has been reconstructed relatively better with the other algorithms. But it should be noted that the best performances were recorded among the ranks of the proposed swarm intelligence algorithms. So, by modeling the object to be reconstructed with supershapes, and by applying the adequate swarm algorithm, we increase our chances of obtaining a better 3D reconstruction.

## 5. Conclusion and Future Works

In this paper, we deal with the problem of three-dimensional reconstruction from a 3D point cloud. The particularity of our approach lies in the modeling of 3D objects by recent parametric shapes, called supershapes and adopting population-based optimization. The use of metaheuristics and especially the family of swarm intelligence algorithms also contributes to the originality of our work. Every detail of our approach has been treated with care and great interest, in particular the choice of the fitness function, and the initialization of the parameters relating to each algorithm. In parallel with the resolution of the stated problem, we have tried to propose an approach presenting robust and optimal results while respecting a reasonable and optimal execution time.

The obtained results exceeded our expectations and are very promising for all tested algorithms. The Artificial Bee Algorithm stands out from the group with its best results. But it should be noted that the other algorithm has enormous potential. Indeed, the results differ from a supershape to another. On the four models of tested supershapes, Artificial Bee Algorithm was more efficient, but perhaps on other different supershapes, Chicken Swarm Algorithm may exceed their current performance and present better results. This hypothesis could guide us in our future work. We can develop a system that allows us to test several existing swarm intelligence algorithms independently on each supershape, then compare and display the best result, obviously respecting the cost constraint. Then, by having a good reconstruction of supershapes, the three-dimensional reconstruction of any 3D object can be carried out first by dividing it into several basic supershapes, and then by grouping the supershapes reconstructions results. We can also diversify our supershapes database by adding different transformation parameters [59] to the initial parameters of the supershape.

## References

- [1] Bernard, A.; Remy, S. (2002): State of the art of the 3d scanning devices. Scanning 2002. Paris, France.
- [2] Gielis, J.; Beirincx B.; Bastiaens E. (2003): Superquadrics with rational and irrational symmetry. Proc. eighth ACM Symp. Solid Model, Appl. SM'03, no. January, p.262.
- [3] Barr, A.H. (1981): Superquadrics and Angle-Preserving Transformations. In IEEE Computer Graphics and Applications.
- [4] Yang, XS (2011): Metaheuristic optimization: Algorithm analysis and open problems. In: Pardalos P.M., Rebennack S. (eds) Experimental algorithms. Lecture Notes in Computer Science, vol 6630. Springer, Berlin, Heidelberg.
- [5] Bonabeau, E.; Dorigo, M.; Theraulaz, G. (1999): Swarm intelligence: from natural to artificial systems. Oxford University Press.
- [6] Laurent, M. (1997): Acquisition au service de l'industrie 3d. Robaut, vol. 16, n°3, pp. 28–32.

- [7] Bernard, A. ; Taillandier, G. (1998): Le prototypage rapide. Hermès.
- [8] Roussel, D. (1999): Reconstruction de courbes et de surfaces 3d en stéréo-acquisition. Phd thesis: University of Paris XI.
- [9] Berger, M. (2016): Problem Books in Mathematics. Springer New York, ISSN 0941-3502, pp 79-84.
- [10] Pentland, A. (1991): Closed-form solutions for physically based shape modeling and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [11] Bajcsy, Solina, F. (1987): Three dimensional object reconstruction revisited. In Proc. of the 1st International Conference on Computer Vision, pp. 231-240, London.
- [12] Sclaroff, S.; Pentland, A.(1991): Generalized implicit functions for computer graphics. Computer Graphics (SIGGRAPH '91 Proceedings), 25(4):247-250.
- [13] Muraki, S.(1991): Volumetric shape description of range data using blobby model. Computer Graphics (SIGGRAPH '91 Proceedings), 25(4):227-235.
- [14] Moore, D.; Warren, J. (1990): Approximation of dense scattered data using algebraic surfaces. TR 90-135, Rice University.
- [15] Vemuri, B. C. (1987): Representation and Recognition of Objects Range Maps. PhD thesis, Department of Electrical and Computer Engineering, University of Texas Austin.
- [16] Schudy, R.B. ; Ballard, D. H.(1978): Model detection of cardiac chambers in ultrasound images. Technical Report 12, Computer Science Department, University of Rochester.
- [17] Schall, O. ; Samozino, M. (2005): Surface from Scattered Points: A Brief Survey of Recent Developments. 1st International Workshop on Semantic Virtual Environments, 138-147.
- [18] Bittar, E.; Tsingos, N.; Gascuel, M.P. (1995): Automatic reconstruction of unstructured 3D data: combining medial axis and implicit surfaces. Computer Graphics Forum.
- [19] Reuter, P.(2003): Reconstruction and rendering of implicit Surfaces from large unorganized point sets. PhD thesis, LaBRI - University of Bordeaux I.
- [20] Turk, G. ; O'Brien, J.F.(2002): Modeling with implicit surfaces that interpolate. In SMI.
- [21] Franke, R.(1982): Scattered Data Interpolation: Tests of Some Method. Mathematics of Computation ,Vol. 38, No. 157, pp. 181-200.
- [22] Ohtake, Y.; Belyaev, A.; Alexa, M. ; Turk, G.; Seidel, H.P. (2003): Multi-level Partition of Unity Implicits. SIGGRAPH 2003, San Diego, California USA, 27-31.
- [23] Tobor, I. ; Reuter, P. ; Schilck, C. (2004): Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. Journal of WSCG 2004, 12:467–474.
- [24] Lancaster, P.; Salkauskas, K. (1981): Surfaces generated by moving least squares methods. Mathematics of Computation 87, 141– 158.
- [25] Levin, D. (1998): The approximation power of moving least-squares. Math. Comput., 67(224):1517– 1531.
- [26] Alexa, M. ; Behr, J. ; Cohen-Or, D. ; Fleishman, S. ; Silva, C.T. (2001): Point set surfaces. IEEE Visualization 2001, pages 21–28.
- [27] Alexa, M. ; Behr, J. ; Cohen-Or, D. ; Fleishman, S. ; Levin, D. ; Silva, C. T. (2003): Computing and rendering point set surfaces. IEEE TVCG, 9(1):3–15.
- [28] Adamson, A. ; Alexa, M. (2003): Approximating and intersecting surfaces from points. Symposium on Geometry Processing 2003, pages 245–254.
- [29] Amenta, N. ; Kil, Y. J. (2004): Defining point-set surfaces. ACM Transactions on Graphics, Proceedings of SIGGRAPH 2004, 23(3):264– 270.
- [30] Boissonnat, J.D. (1984): Geometric structures for three-dimensional shape representation. ACM Trans. Graph.,3(4):266-286.
- [31] Amenta, N.; Bern, M. (1999): Surface reconstruction by voronoi filtering. Discr. Comput. Geom., 22:481– 504.
- [32] Amenta, N. ; Choi, S.; Dey, T. K. ; Leekha, N. (2002): A simple algorithm for homeomorphic surface reconstruction. Internat. J. Comput. Geom. & Appl., 12:125–141.
- [33] Amenta, N.; Choi, S.; Kolluri, R. (2001): The power crust. In Proceedings of 6th ACM Symposium on Solid Modeling, pages 249–260.
- [34] Dey, T. K. ; Goswami, S. (2004): Provable surface reconstruction from noisy samples. In Proc. 20th ACM Sympos. Comput. Geom.
- [35] Hoppe, H. ; DeRose, T. ; Duchamp, T. ; McDonald, J. ; Stuetzle, W. (1992): Surface reconstruction from unorganized points. Computer Graphics, 26(2) :71-78.
- [36] Hoppe, H. ; DeRose, T. ; Duchamp, T. ; McDonald, J. ; Stuetzle, W. (1993): Mesh optimization. Computer Graphics, 27:19-26.
- [37] Boissonnat, J.D. ; Cazals, F. (2000): Smooth surface reconstruction via natural neighbour interpolation of distance functions. In Proceedings of the sixteenth annual symposium on Computational geometry, 223-232.
- [38] Kass, M. ; Witkin, A. ,Terzopoulos, D. (1988): Snakes : Active contour models. International Journal of Computer Vision, 1 :321–332.
- [39] Qin, H. ; Duan, Y. (2001): Intelligent Balloon: a subdivision-based deformable model for surface reconstruction of arbitrary, Unknown Topology. In Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, 47-58.
- [40] Baader, A. ; Hirzinger, G. (1994): A Self-organizing algorithm for multisensory surface reconstruction. In International Conf. on Robotics and Intelligent Systems IROS'94, Munich, Germany.
- [41] Kohonen, T.(1982): Self-organized formation of topologically correct feature maps. In Biological Cyber- netics 43, pages 59–69.
- [42] Fritzke, B.(1995): A Growing Neural Gas Network learns topologies. Advances in Neural Information Processing Systems 7, MIT Press, 625-632.
- [43] Ivrisimtzis ,I.; Jeong, W.K. ; Seidel, H.S. (2003): Using growing cell structures for surface reconstruction, Shape Modeling. International (SMI 2003), Seoul, Korea, IEEE, Los Alamitos, 78-86.
- [44] Dréo, J. ; Petrowski, A. ; Taillard, É. D. ; Siarry, P. (2003): Métaheuristiques pour l'optimisation difficile. Eyrolles (Editions), ISBN 2212113684.
- [45] Cavazzuti, M.(2013): Deterministic Optimization. In: Optimization Methods. Springer, Berlin, Heidelberg.
- [46] Marti,K. (2005): Stochastic Optimization Methods Application in Engineering and Operations Research. Third Edition Springer, pp 1- 35.
- [47] Sadgal, M.; Abdelaziz, A.; Mansouri,F. (2016): A Discrete Particle Swarm Optimization to Estimate Parameters in Vision Tasks. IJACSA, 7(1) 196–207.
- [48] Benchikhi, L. ; Sadgal, M. ; Elfazziki, A. ; Mansouri, F. (2017) : An ant colony based model to optimize parameters in industrial vision, ELCVIA , Vol 16, No 1 (2017) 1577-5097.
- [49] Mansouri, F. ; Benchikhi, L. ; Sadgal, M. ; Elfazziki, A. (2017): A Combined Cuckoo Search Algorithm and Genetic Algorithm for Parameter Optimization in Computer Vision (IAER), vol 12, no 22, pp. 12940-12954.
- [50] Karaboga, D. ; Basturg, B. (2007): A powerful and efficient algorithm for numerical function optimization : artificial bee colony (ABC) algorithm. Journal of Global Optimization, vol 39, no. 3, pp.459-471.

- [51] Meng, X. ; Liu, Y.; Gao, X. ; Zhang, H. (2014): A New Bio-inspired Algorithm: Chicken Swarm Optimization. In: Tan Y., Shi Y., Coello C.A.C. (eds) *Advances in Swarm Intelligence. ICSI 2014. Lecture Notes in Computer Science*, vol 8794. Springer, Cham.
- [52] Barr, A. H. (1984): Global and local deformations of solid primitives. *Comput. Graphics*, vol. 18, no. 3, pp. 21-30.
- [53] Gross, A.D. ; Boust, T.E. (1988): Error of Fit Measures for Recovering Parametric Solids. In *Proceedings of the Second International Conference in Computer Vision*, Tampa, Florida, USA, pp. 690–694.
- [54] Zhang, Y. (2003): Experimental comparison of superquadric fitting objective functions. *Pattern Recognition Letters*, vol. 24, pp. 2185–2193.
- [55] Fougerolle, Y. D. ; Gribok, A.; Foufou, S. ; Truchetet, F. ; Abidi, M.A. (2005a): Boolean operations with implicit and parametric representation of primitives using R- functions. *IEEE Transactions on Visualization and Computer Graphics*, 11(5) :529–539.
- [56] Fougerolle, Y. D. ; Gribok, A. ; Foufou, S. ; Truchetet, F. ; Abidi, M. A. (2005): Implicit surface modeling using supershapes and R-functions. In *Proc. Pacific Graphics'05*, pp. 169–172, Macao, China.
- [57] Bokhabrine, Y. ; Fougerolle, Y. D. ; Foufou, S. ; Truchetet, F. (2007): Genetic algorithms for gielis surface recovery from 3D data sets. In *Proceedings -International Conference on Image Processing, ICIP*.
- [58] Remmach, H. ; Sadgal, M. (2019): 3D Object Reconstruction From 3D Point Cloud By Supershapes Using PSO. *Journal of Theoretical and Applied Information Technology*, Vol.97. No 24.
- [59] Wyvill, B. ; Guy, A. ; Galin, E. (1999): Extending the CSG tree. Warping, blending and Boolean operations in an implicit surface modeling system. *Comput. Graph. Forum*, vol. 18, no. 2, pp. 149–158.