A Discrete Particle Swarm Optimization to Estimate Parameters in Vision Tasks

Benchikhi Loubna
Department of computer science
Computer Systems Engineering Laboratory
Marrakesh, Morocco

Elfazziki Abdelaziz
Department of computer science
Computer Systems Engineering Laboratory
Marrakesh, Morocco

Sadgal Mohamed
Department of computer science
Computer Systems Engineering Laboratory
Marrakesh, Morocco

Mansouri Fatimaezzahra
Department of computer science
Computer Systems Engineering Laboratory
Marrakesh, Morocco

***Abstract*—The majority of manufacturers demand increasingly powerful vision systems for quality control. To have good outcomes, the installation requires an effort in the vision system tuning, for both hardware and software. As time and accuracy are important, actors are oriented to automate parameter's adjustment optimization at least in image processing. This paper suggests an approach based on discrete particle swarm optimization (DPSO) that automates software setting and provides optimal parameters for industrial vision applications. A novel update functions for our DPSO definition are suggested.**

The proposed method is applied on some real examples of quality control to validate its feasibility and efficiency, which shows that the new DPSO model furnishes promising results.

Keywords—component; component; industrial vision; image processing; optimization; DPSO; quality control

I. INTRODUCTION

Mostly, in a vision system, the problem of quality control requires a very high precision in object extraction often under imposed constraints. A variety of methods have been developed since decades towards image processing in Quality Control (QC), but the problem of operators' choice and their parameters' setting is still relevant, these settings are usually made by trial and error. In fact, algorithms may be adopted after a long series of tests.

The choice of operators and their parameters' adjustment require specific knowledge, and must be done experimentally due to the lack of automatic mechanisms. In spite of vision systems' diversity and the rich library of image processing approaches scientifically strong, the user intervention in most applications remains necessary.

Few systems have succeeded in automating vision applications without requiring user's intuition. B.Nikolay and B.Schneider and S.Jacob [1] proposed a method to optimize automatically parameters of vision systems for surface inspection. Their method is based on evolutionary algorithms. The major types that have been used are: Evolution strategies (ESs) and Evolutionary Programming (EP).

Several studies have been done and few authors proposed methods such as numerical optimization, which apply mathematical or statistical techniques to minimize, or maximize, an objective function defined over a parameter space. Few years later, Taylor proposed a method based on reinforcement learning to monitor parameters in vision applications [2]. However, different techniques have been developed on the basis of population approaches: Genetic Algorithms [3], Ant Colony Optimization (ACO) [4] and Particle Swarm Optimization [5]. In such approaches, each individual (agent) in a population start by building an approximate solution. With a mechanism of interaction and evolution, individuals converge towards the optimal solution.

Most of the proposed techniques have not been widely adopted for the parameter-tuning problem. This can be partly awarded to few application examples in the real image analysis. Visually, with different parameter types and the variety of their values, the problem is NP-complex.

The objective is to deploy the artificial intelligence (AI) techniques [6] for solving such problems. The most important property in distributed AI is the cooperation between agents to provide the best solution. In DPSO, particles are considered as agents that cooperate in some way to reach this goal.

The contribution of this paper is a novel model using DPSO adapted to parameters' adjustment. In most cases, the study of a vision operator concerns a discrete domain of parameters' values; so, the study is focalized on the standard PSO reformulation in a discrete way. Optimal parameter values for applications in vision systems are found. It is shown that using a discrete PSO is more adapted to parameter tuning and allows achieving good quality results. The approach is applied on real image examples in a quality control task and the outputs are compared to references.

The reminder of this paper is organized as follows, Section 2 gives an over view on the use of image processing in quality control. Section 3 describes the proposed approach. The experimental results are presented in Section 4. Finally conclusions are stated in Section 5.

II. State of the Art

In the majority of vision tasks, the user is required and sometimes obliged to combine several operators, each one has a multitude of parameters to be adjusted, but few systems have succeeded in automating vision applications without requiring user's intuition. Some authors searched to automate completely the process, S.Treuill, D. Driouchi and P.Ribereau [7] proposed a method to adjust parameters in an image processing chain based on an experimental 2^{k-p} factorial plan applied to a vision system designed for measuring the neck ratio of a sugar beet batch. Recently L.Franek and X.jiang [8] proposed to use orthogonal plans of experiments for parameter learning. They analyze means to estimate the optimal parameter setting. In addition, a combination of orthogonal arrays and genetic algorithm is used to further improve the performance.

A different technique was introduced in [9], using interactive visualization to develop novel histopathology image segmentation software, which illustrates its potential usefulness for parameter optimization purposes. In recent years, parameters' optimization in image processing is supported by artificial intelligence techniques [10], such as multi-agent architecture. I.Qaffo, M.Sadgal and A.Elfaziki [11] proposed an automatic method based on reinforcement learning for object recognition, using two types of agents: User Agent (UA) and Parameter Agent (PA). The UA gives necessary information to the system, as the combination of applicable operators, the set of adjustable parameters for each operator, and a values' range for each parameter. Then, it generates a PA for each combination of operators. The PA uses reinforcement learning to assign the optimal values for each parameter in order to extract the object of interest from an image. One of the most used methods in parameter optimization in image processing is population-based heuristics. Genetic algorithm [12] is widely used for this purpose. We proposed an optimization method based on the standard particle swarm optimization [13] to find the best values of free algorithm parameters used in image processing. The method is restricted to operators with numerical parameters and continuous fields.

On the other hand, particle swarm optimization (PSO) was first introduced by R. Eberhart and J. Kennedy [14] in 1995, as a novel nature inspired method from social behavior of birds in a flock. It is used on optimization problems that are partially irregular, noisy, change over time, etc.

Since 2002, researches applying PSO has grown rapidly. Popularity of PSO is due to its several strengths namely: very few parameters to adjust, its easiness of implementation, its robustness, and its convergence speed.

Many problems used this paradigm, like combinatorial optimization problems including vehicle routing problem [15], traveling salesman problem [16], and scheduling problems [17] [18].

Other application areas of PSO, the most potential includes fuzzy controller design for mobile robots [19], recognition of control chart patterns [20], real time robot path planning [21], image segmentation [22], speaker verification [23] and gesture recognition [24], to name a few.

All these researches have proved the PSO efficiency as a new tool to obtain satisfying optimization results. This study adopts a discrete PSO to present a new model optimizing vision systems operators' and parameters' values. This method proves its feasibility and efficiency.

III. Formulation of the Optimizition Problem

A. Vision systems and tasks

Vision systems are designed to produce applications in image processing, object recognition, etc. Such applications come with several tasks and operators that transform a product stream into an information flow. A vision system is built on a succession of tasks (some ones can be executed in parallel), which can be a set of other tasks or an elementary task composed of different phases of treatment "Fig. 1''.

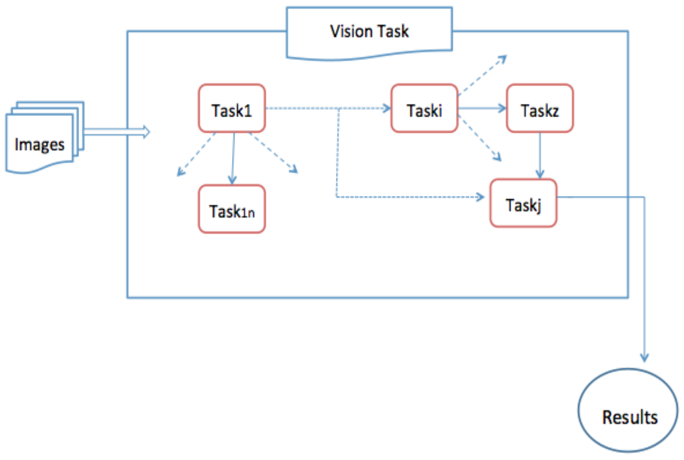


Fig. 1. General process to implement vision systems

Image processing used in the quality control domain, consists of supervising the basic parameters describing an image quality, in order to extract necessary measurements for control. The main problem here remains the segmentation quality, which affects hardly controls.

Image processing helps to increase flexibility and productivity in production factories. Further, it takes a hand in maintenance and enhances knowledge about the quality of products. Furthermore, it offers the advantage of being able to interfere at several levels, for example: the real-time quality inspection of gelatin capsules in pharmaceutical applications [25]. The designed image processing system is based on some tasks illustrated in ''Fig. 2''. Edge-based image segmentation technique for quality inspection is used to detect accepted and rejected capsules.

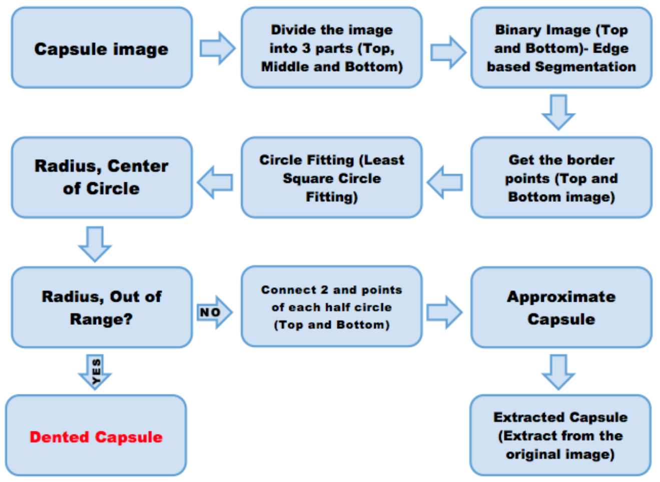

Capsule image
Divide the image into 3 parts (Top, Middle and Bottom)
Binary Image (Top and Bottom)- Edge based Segmentation
Get the border points (Top and Bottom image)
Circle Fitting (Least Square Circle Fitting)
Radius, Center of Circle
Radius, Out of Range?
NO
Connect 2 and points of each half circle (Top and Bottom)
Approximate Capsule
YES
Dented Capsule
Extracted Capsule (Extract from the original image)

Fig. 2. Flow chart of the proposed capsule extraction method (source [25])

IV. Operators and Parameters

To accomplish a vision task, it is necessary to go through multiple tasks; each one is a succession of several phases. Each phase has a set of possible operators.

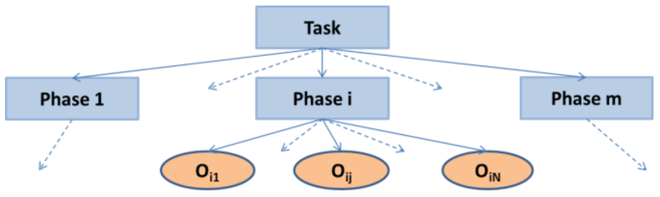


Fig. 3. Possible operators for each phase in a vision task

We may have several combinations of these operators, each combination can achieve the considered task, but the output is qualitatively different. Let's consider a task made of several phases. Since each phase concocts a set of feasible operators, n different operator's combinations C_k are built.

$$C_k = (O_1, O_2, ..., O_N) \qquad k=1....n \tag{1}$$

An operator (O_i) requires fixing some parameters; a range of possible values for each parameter is given out ''Fig. 4''.

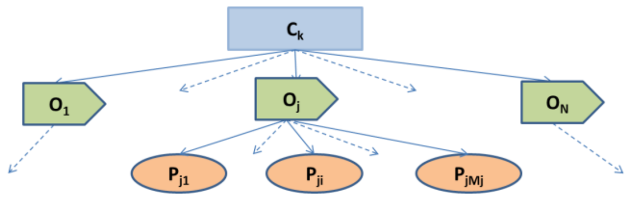


Fig. 4. Operator's combination and their parameters

The vision task performance is based on these parameter settings, and their variation strongly influences results. Below we mention some adjustable parameters of different algorithms such as edge detection.

When applying an edge operator, which identify points in a digital image at which the image brightness has discontinuities, a lot of filters can be applied here such as sobel, prewit and log, each one of these filters has a free parameter: the threshold, to remove edges with poor contrast and then contours are formed by pixels higher than a given threshold. ''Fig. 5'' illustrates a segmentation done with different threshold values for a chosen filter (canny).

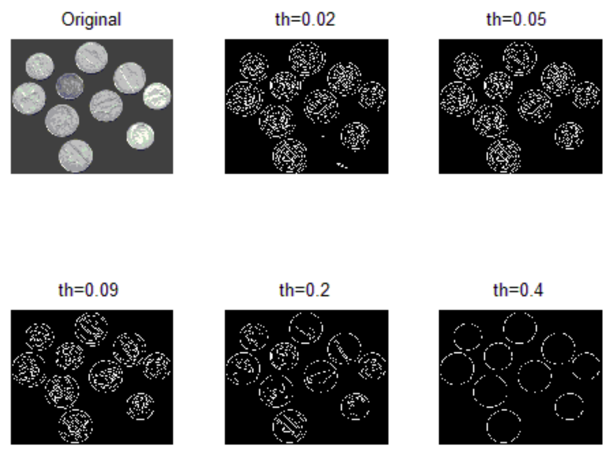


Fig. 5. Applying a canny filter with different thresholds values

A. *Objective function definition*

Mathematically the problem can be defined as follows: A model P = (D, ω, Err) where:

- D is the solution space, defined over a set of variables, and ω a set of constraints among the variables.
- An objective function Err: D → R^+ to be minimized.

A solution d ϵ D is a complete assignment in which each variable has an assigned value; d must satisfy all the constraints. A feasible solution d* is a global optimum if:

$$Err(d^*) < Err(d) \qquad \vee\, d \in D \tag{2}$$

Back to our problem, a combination of operators C_k and an input image I_{input} are considered. The application of C_k provides an output result R_k (image or features):

$$R_k = C_k(I_{input}) \tag{3}$$

To evaluate the result, an objective function is necessary to compare qualitatively the outputs (R_k), a reference led by an expert is used as a ground truth R_r. The rating calculated by this objective function represents the error between the two images.

$$Err_k(C_k) = fitness(R_k, R_r) \tag{4}$$

The fitness function definition depends on the vision task. For sure optimal parameters' values give the best output. Consequently, this corresponds to the smallest error (*Err*), and then the operator's combination to adopt is the one corresponding to the minimal error:

$$Err = Min(Err_k(C_k)) \tag{5}$$

k=1… n

B. *State definition*

Let's consider an operator's combination applied to the input image I_{input}, based on ''Fig. 4'', (1) and (2) we note:

$$I_{output} = (O_1, O_2, \ldots, O_N)(I_{input}) \tag{6}$$

I_{output} is the result of applying operators' combination on the input image I_{input}.

To simplify, we consider a combination as a sequence of N operators where each operator is applied over the output of its predecessor:

$$I_j = O_j(I_{j-1}) \text{ for } j=1 \dots N,\ I_{input} = I_0,\ I_{output}=I_N=R_N \quad (7)$$

Considering that each operator O_j has M_j parameters and $M = \sum_{j=1}^{N} M_j$ is the number of all parameters. The error is simply function of parameters denoted by:

$$Err(p_1, p_2, \dots, p_M) = fitness\ (R_N, R_r) \quad (8)$$

Where p_i is a parameter taking values in a domain D_i. Then, we consider the cartesian product $D=D_1*D_2*\dots*D_M$, and we call a state of parameters' values each M-uplet $(u_1, u_2, \dots, u_M) \in D$ where u_i is a value of parameter p_i.

Using these notations, the problem is to find a state (an M-uplet) that minimizes the error function (Err) over the domain D.

So, the solution is the state $(u_1^*, u_2^*, \dots, u_M^*) \in D$, such as $Err(u_1^*, u_2^*, \dots, u_M^*)$ is minimal or:

$$(u_1^*, u_2^*, \dots, u_M^*) = ArgMin(Err(u_1, u_2, \dots, u_M)) \quad (9)$$

The objective function is not expressed directly with parameters, but it is established on the basis of results. In fact, direct numerical methods could not be applied to solve this problem. In contrast, the proposed method belongs to relaxation methods, which are preferred to solve this sort of problems. Our approach consists of searching to converge toward minimal error in an iterative way, relying on an optimization model.

V. The Proposed Approach

As mentioned above, there is no model to establish directly a relationship between objective function and parameters, so direct numerical methods could not be applied to solve this problem. Following, we describe the procedure we employed to solve this problem applying a discrete PSO algorithm.

A. *The PSO Model*

The particle swarm optimization is based on the social behavior reproduction developed by R.Eberhart and J. Kennedy [14].

Particle swarm optimization is a population-based optimization algorithm modeled after the simulation of social behavior of birds in a flock. It is based on a set of individuals randomly arranged, called particles moving in the search space, each one is a potential solution and the aim is to get closer to the best solution.

A particle swarm is characterized by:

- The number of particles in the swarm.
- The topology and the neighborhood size of a particle that define his social network.
- The inertia weight of a particle, denoted w.
- The confidence factors, denoted by r_1 and r_2, which weigh the tendency to return towards the best solution visited and the tendency to follow the neighborhood.

The performance of each particle, i.e. how close the particle is from the global optimum, is measured using an error function called also fitness function, which depends on the optimization problem.

Each particle has a memory about his best visited solution, as well as the ability to communicate with the particles forming his entourage. From this information, each particle keeps informed of its location and ability (the optimized function value), as well as the best place, and its corresponding ability, she has met so far in its flight.

Each particle i fly through an n–dimensional search space, and maintain the following information:

- x_i, the current position.
- $pbest_i$, the personal best position.
- v_i, the current velocity.

Building an m-uplet from predefined domains D_i forms the position of each particle. Then, the objective function evaluates the applied operators result with the selected parameters' values.

A general model of a particle swarm optimization algorithm is presented as:

```
Procedure PSO
Initialize a population of particles
do
    for each particle i with position $x_i$ do
        if ($x_i$ is better than $pbest_i$) then
            $pbest_i \leftarrow x_i$
        end_if
    end_for
    Define $gbest_i$ as the best position found so far
    by any of particles' neighbors
    for each particle do
        $v_i \leftarrow$ update_velocity($x_i$, $pbest_i$, $gbest_i$)
        $x_i \leftarrow$ update_ position($x_i$, $v_i$)
    end_for
While (a stop criterion is not satisfied)
```

In Standard PSO, domains are continues and the velocity updates are calculated as a linear combination of position and velocity vectors. Thus, a particle velocity is updated using (10) and the position of this particle is updated using (11).

$$v_i(t+1)=w * v_i(t) + r_1 * (p_i(t) - x_i(t)) + r_2 * (gbest - x_i(t)) \quad (10)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (11)$$

In the formula, w represents the inertia weight [14], $gbest$ is the best position among the best previous positions of particle informants. r_1, r_2 are numbers from a random distribution, and v_i must be in the range $[-V_{max}, V_{max}]$, where V_{max} is the maximum velocity.

This process is iterative and parallel, where in each iteration, all particles update their positions and must stop if there is a convergence, or after a fixed number of iterations.

B. Description of the proposed discrete PSO (DPSO)

In this case, parameter domains are discrete and some are non-digital. Obviously we cannot use directly the standard version of PSO designed for continuous domains. Several studies on particular applications, have discrete formulations of PSO (DSPO), beginning with BPSO (binary version) [26] to multivariate problems [27].

For discrete optimization problem, conventional PSO algorithm must address the following two issues:

- How to change the position of a particle?
- How to guarantee that positions are reasonable?

In DPSO, each particle represents its position in binary values, 0 or 1. Each particle's value can then be changed, or better say mutate, from one to zero or vice versa. In DPSO particle's velocity is defined as the probability that this particle changes its state to one [28].

The particles move in a state space restricted to 0 and 1, with a certain probability depending on individual and social factors. The probability of $x_i(t) = 1$, $Pr(x_i = 1)$, is a function of $x_i(t\text{-}1)$, $v_i(t\text{-}1)$, $pbest_i(t\text{-}1)$ and $gbest_i(t\text{-}1)$.

The probability of $x_i(t) = 0$ equals $1 - Pr(x_i = 1)$. Thus (10) is replaced by (12), where *rand*3 is a random number, sig($v_i(t)$) is a logic transformation which can constrain $v_i(t)$ to the interval [0,1] and can be considered as a probability:

$$sig(v_i(t)) = \frac{1}{1+e^{-v_i(t)}} \qquad (12)$$

$$x_i(t) = \begin{cases} 1 & if\ rand3 < sig(v_i(t)) \\ 0 & otherwise \end{cases} \qquad (13)$$

To extend the idea, another approach is proposed by M.Clerc [29] using the Traveling Salesman Problem (TSP) to illustrate the PSO concept for discrete optimization problems. M.Clerc defines a domain as a set of states with an order on objective functions values. He presents also some operations with position and velocity such as addition, subtraction and multiplication specific to TSP. A distance is defined to be utilized with physical neighborhoods. The idea is interesting, but the problem's definition and update functions depend on the application.

Inspired by M. Clerc idea regarding the states' set, we propose a discrete PSO model based on following definitions:

- $D=D_1*D_2*\ldots*D_M$ is considered as a set of states (or a search space).

Each M-uplet x = (u_1, u_2, …, u_M) is a state.

- Objective function Err is discrete and numerical with an order on states:

Err(x)>= Err(x') or Err(x') >= Err(x)

- Position of a Particle is a state
- Domain representation: each value in the domain can be represented by its relative location. These locations are fixed in some conventions.

The model is illustrated in ''Fig. 6''.

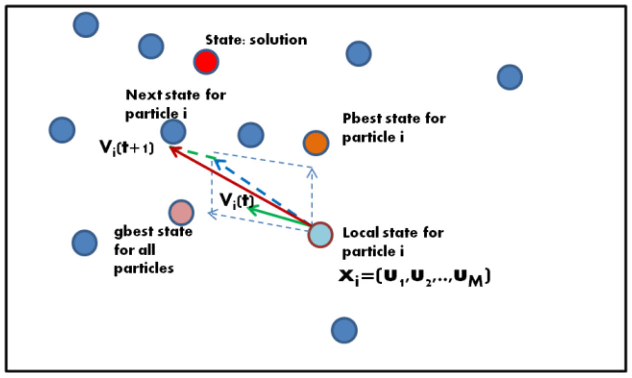

State: solution
Next state for particle i
Pbest state for particle i
$V_i(t+1)$
$V_i(t)$
gbest state for all particles
Local state for particle i
$x_i=(u_1,u_2,\ldots,u_M)$

Fig. 6. Space State Presentation

A domain representation can be expressed in many ways to facilitate operator's definition on states. One of these representations is given here:

A domain $D_j=\{u_{1j},\ldots,u_{kj},\ldots,\ u_{nj}\}$ is ordered from 1 to n, n=$|D_j|$.

Each value has a location in the domain and then two reciprocal functions *Rank* and *Value are* defined:

$Rank(u_{kj})=k$, the rank of a value u_{kj} in D_j , $(k=1,\ldots n)$

$u_{kj} = Value(k)$, the value in D_j of rank k.

Example:

$D_j=\{a,b,c,d\}$
$Rank(a)=1,\ \ldots,\ Rank(d)=4$
$Value(1)=a,\ \ldots, Value(4)=d$

The velocity can be expressed as a distance between ranks of two separated states.

C. The update functions

Since the DPSO problem claims to define positions discretely and updates them using velocity, we describe bellow the new proposition.

a) Add/subtract operators: To add two values at rank *k* and *k'* we simply move to the rank *k+k'* and the result is the value corresponding to rank *k+k'*. Since D_j has a limited size $|D_j|$, *k+k'* is calculated modulo $|D_j|$. Then the addition operator can be defined as:

$u_{kj} \oplus u_{k'j} = u_{(k+k')j}$

Idem for Subtraction, the result correspond to *k-k'* modulo $|D_j|$:

$u_{kj} \ominus u_{k'j} = u_{(k-k')j}$

The position of a particle is a state $x = (u_1,u_2,..u_M)$, were u_j represent values in D_j. To change the state we extend operations:

$x\oplus x'=(u_1\oplus u'_1,u_2\oplus u'_2,..,u_M\oplus u'_M),$

$x \ominus x' = (u_1 \ominus u'_1, u_2 \ominus u'_2, .., u_M \ominus u'_M)$,

$Rank(x) = (Rank(u_1), \ldots, Rank(u_M))$,

$x = (Value(k_1), \ldots, Value(k_M))$, were k_j represent ranks of values in D_j.

b) Velocity update: We use the same expression in (10) with the new add/subtract operators and the velocity is expressed as a vector of moves (left: - and right: +)

$$v(t+1) = round(w * v(t) + r_1 * (Rank(p(t)) - Rank(x(t))) + r_2 * (Rank(gbest) - Rank(x(t)))) \text{ modulo } (|D_1|, \ldots, |D_M|) \quad (14)$$

c) Position update: To update the value of a parameter, we update at first its rank:

$$Rank(x(t+1)) = Rank(x(t)) + v(t+1) \text{ modulo } (|D_1|, \ldots, |D_M|) \quad (15)$$

Then, we obtain the correspondent value:

$$x(t+1) = Value(Rank(x(t+1))) \quad (16)$$

VI. Application

In the previous section the novel DPSO approach is described, the problem of choosing optimal operators and the optimal values of their parameters in a vision task is solved. The approach we presented in theory is applicable to any vision task that needs operators' selection or parameters' adjustment or both of them. In this section our approach is tested on two different tasks of image processing, first one is about contours detection of mechanical objects and second one is about text recognition and aspect inspection related to tickets label on industrial products.

A. Case study 1: Contour detection of mechanical objects.

a) Phases, operators and parameters determination

The task of contours detection allows identifying areas, of a digital image, corresponding to a brutal change in light intensity. It significantly reduces data quantity and eliminates the information judged less relevant, while preserving the important structural properties of the image, in order to extract measurements for example. This task is made of tree phases of treatment ''Fig. 7'', firstly a pre-processing phase is necessary to remove any noise from the image, and then processing phase would take place to determine object contours, a post-processing phase will go after to eliminate insignificant contours. A comparison between a segmentation done by experts in image processing domain, and results obtained by DPSO approach will be done.

- Pre-processing phase

Preprocessing phase consists of improving image quality using filters; we propose a list of tree filters predefined in Matlab: medfilt2, ordfilt2 and wiener2. Medfilt2 is a nonlinear operator called median filtering, used in image processing to reduce "salt and pepper" noise, ordfilt2 is also a nonlinear operator it is an order-statistics filtering, it replaces each element in the image by the orderth element in the sorted set of neighbors specified by the non zero elements in domain. Finally wiener2 is an adaptive noise-removal filtering; it uses a pixel wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.

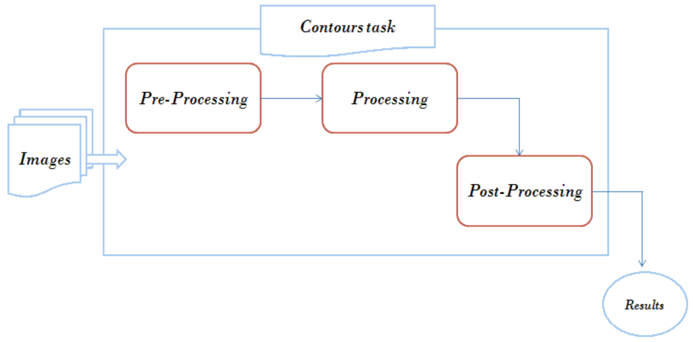


Fig. 7. Treatment phases of a contour task

To apply these filters a parameter must be specified: the filter size called t_w, a range of possible values would be provided ''Fig. 8''.

- Processing phase

Processing phase consists of detecting edges (contours), the operator applied is predefined in Matlab: edge[1] with two parameters to be adjusted, filter to use and the threshold. A range of possible values would be provided for each parameter ''Fig. 8''.

- Post-processing phase

Post-processing phase consists of refining the image by deleting small objects. The operator 'bwareaopen' would be applied as one operator of this phase; it is a morphological operator, which removes from a binary image all objects that have connectivity inferior than a predefined threshold. A range of possible thresholds and connectivity values would be provided ''Fig. 8''.

- Objective function

The objective function, called also error function, depends on the optimization problem. In this case, contours detection task, many adapted error functions are possible. The error of a particle should indicate how good the segmentation of the input image is, in comparison to the target segmentation.

The error function used here is based on the confusion matrix for a two-class classifier. Several standard terms have been defined for the two-class matrix; the one used in this work is the accuracy, which represents the proportion of the total number of predictions that were correct. It is determined using the equation:

$$Acc = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad (17)$$

Where t_p (true positive) represent white pixels well ranked (contours), t_n (true negative) represent dark pixels well ranked, f_p (false positive) represent contours misclassified a [1] nd f_n (false negative) represent dark pixels misclassified.

[1] http://www.mathworks.com/help/images/ref/edge.html

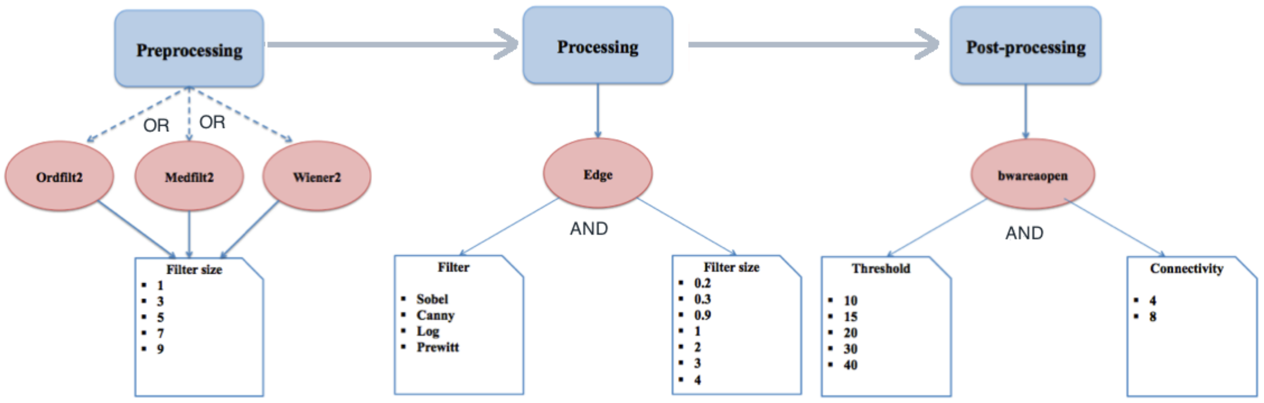


Fig. 8. Operators' and their parameters possible values for each phase

In reality, contour pixels found are compared to those in the ground truth image using DPSO approach. The error function considered is:

$$Err= 1- Acc \tag{18}$$

Applying DPSO approach associates to each operator's combination an error value, in addition to its best parameter values.

$C_1 (O_1(p_2), \ldots, O_n(p_k)) \dashrightarrow Err_1$

$C_2 (O_1(p_3), \ldots, O_n(p_j)) \dashrightarrow Err_2$

.

.

.

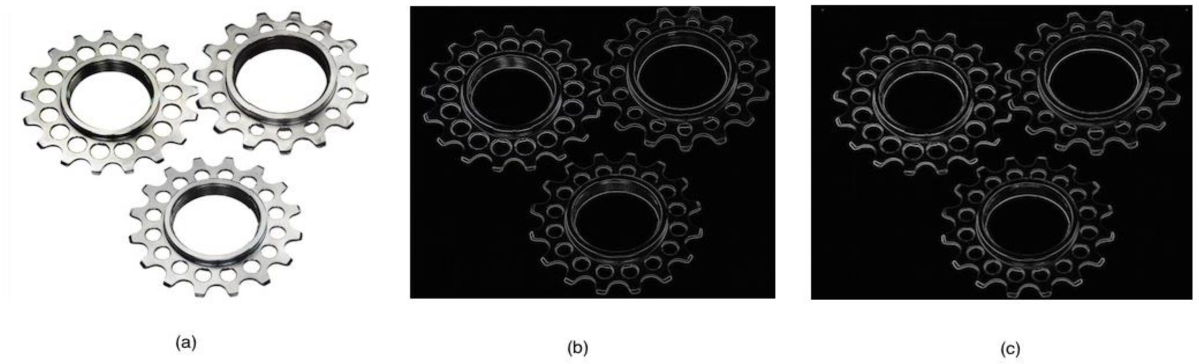
$C_n (O_1(p_1), \ldots, O_n(p_m)) \dashrightarrow Err_n$

The best combination is as: min (Err_1, Err_2, …. , Err_n).

b) Results and Discussion: The experiment was conducted on a dataset of mechanical objects images. DPSO model is applied and table 1 resumes best parameter's values and error rates of best operator's combination, and a simple of images' result is shown in ''Fig.9''.

TABLE I. BEST OPERATORS COMBINATION AND THEIR BEST PARAMETERS' VALUES FOR TEST IMAGES

		Pre-processing	Processing		Post-processing		Error rate
Image 1	**Operator**	Medfilt2	Edge		Bewareaopen		0.0023
	Parameters	1	Sobel	2	30	8	
Image 2	**Operator**	Medfilt2	Edge		Bewareaopen		0.0041
	Parameters	3	Prewitt	3	10	8	
Image 3	**Operator**	Ordfilt2	Edge		Bewareaopen		0.0012
	Parameters	2	Sobel	0.9	15	8	



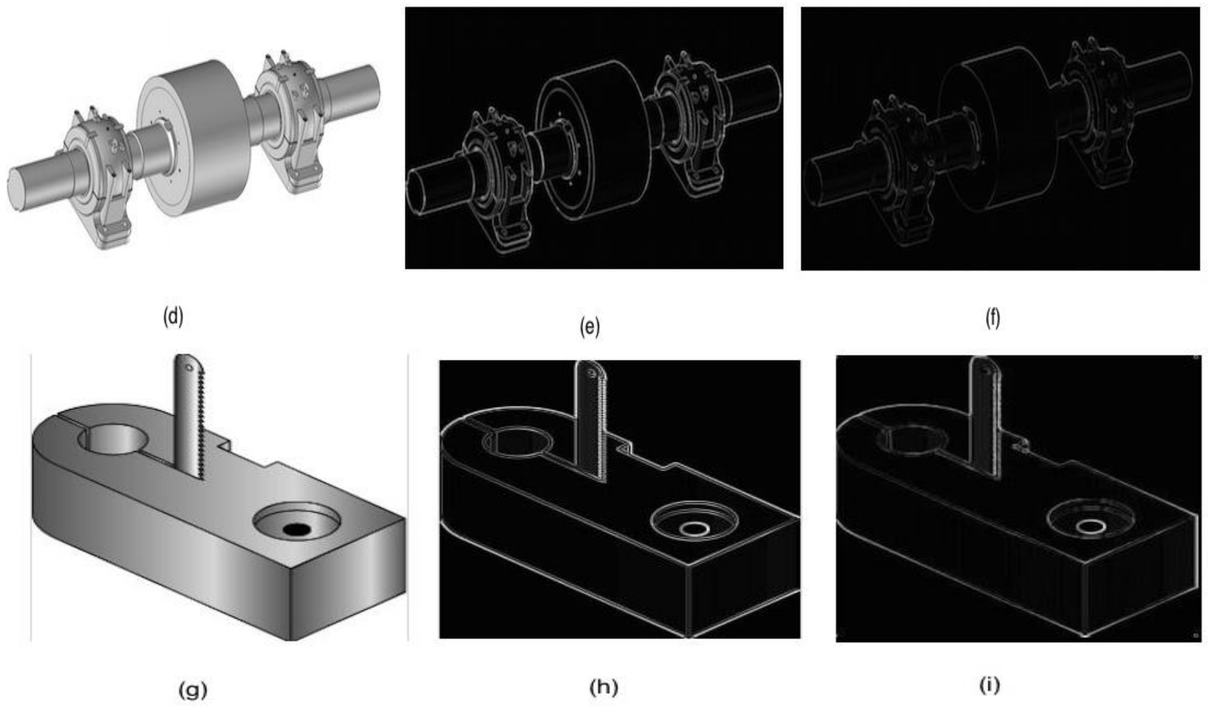


Fig. 9. From left to right: image of mechanical object, segmentation done by an expert and finally DPSO approach result

The proposed method achieves good results with a small error rate, but to improve even more the quality of results we will try in the following to adjust parameters of DPSO algorithm.

B. Case study 2: Text recognition

Let's consider a line producing bottles, a vision system will be configured, in order to control the stickers presence on the bottles, verify its position, the presence of the logo and the trade name.

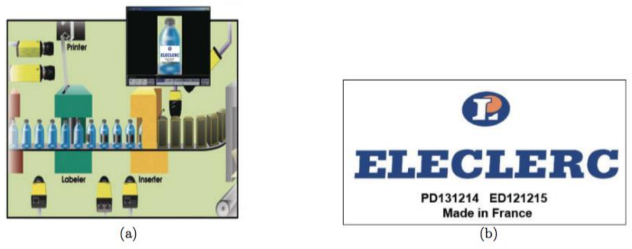


Fig. 10. Production line of bottles using a vision system for quality control. (b) The sticker image

The process of aspect inspection consists of detecting the ticket presence and making sure that it contains the correct logo and content. ‘‘Fig. 11’’ shows different phases performed to achieve this vision task and illustrate operators used in each phase.

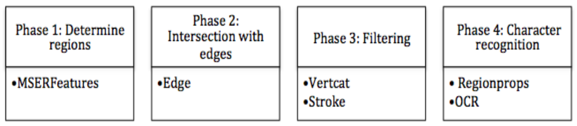

Phase 1: Determine regions
•MSERFeatures
Phase 2: Intersection with edges
•Edge
Phase 3: Filtering
•Vertcat
•Stroke
Phase 4: Character recognition
• Regionprops
•OCR

Fig. 11. Different phases of the vision task and operators candidates for each phase

Each one of these operators, except OCR, has some parameters to be fixed depending on the image we are working on. To find out the parameter values optimal combination, a range of values for each one of parameters is given up; a list of possible values of each parameter is provided in table 2.

TABLE II. VALUES CANDIDATES FOR EACH PARAMETER

Operators	Mserfeatures		Edge		Vertcat		Strock	RegionProps
	Max	Min	Threshold	Filter	Eccentricity	Solidity	Width variation	Areathreshold
Values candidates for each parameter	10 20 30 40 50 60	2000 3000 4000 5000 6000 7000	0,001 0,002 0,003 0,004 0,005 0,006	Canny Log Prewitt Sobel	0,970 0,975 0,980 0,985 0,990 0,995	0,1 0,2 0,3 0,4 0,5 0,6 0,7	0,3 0,4 0,5 0,6 0,7 0,8 0,9	10 20 30 40 50 60 70

Fixing parameters for an optimization algorithm is a very important step before proceeding to parameter optimization of a vision task. DPSO is applied using 30 particles; inertia weight is fixed to 0.5, Confidence factor r_1 to 0.3 and Confidence factor r_2 to 0.7. We run the algorithm for 30 iterations, for the only operator's combination of this task:

[MserFeatures, Edge, Vartcat, Strock, Regionprops, OCR]

The objective function used to evaluate this task is composed of three steps, since we have to found three different regions: the first one must contain the logo, and it is compared to its reference using corr2, a predefined function in Matlab returning the correlation coefficient between two images. Second and third ones must contain a text, which we compare to references and penalize each unrecognized letter. The optimal parameter values found are summarized in table 3, and ‘‘Fig. 12’’ demonstrates experimental results of each operator, using optimal parameters. Three different regions are detected. First one, which contains the logo, is compared to its reference. The second and third regions contains a text, which we recognize using OCR and compare to references. The final error rate represents the sum of the three error rates calculated over the three text regions found.

TABLE III. OPTIMAL PARAMETERS VALUES

Parameter	**Optimal Value**
Area-Range	[10, 6000]
Filter	Canny
Threshold	0,04
Eccentricity	0,995
Solidity	0,2
Width-variation	0,9
Area-threshold	10

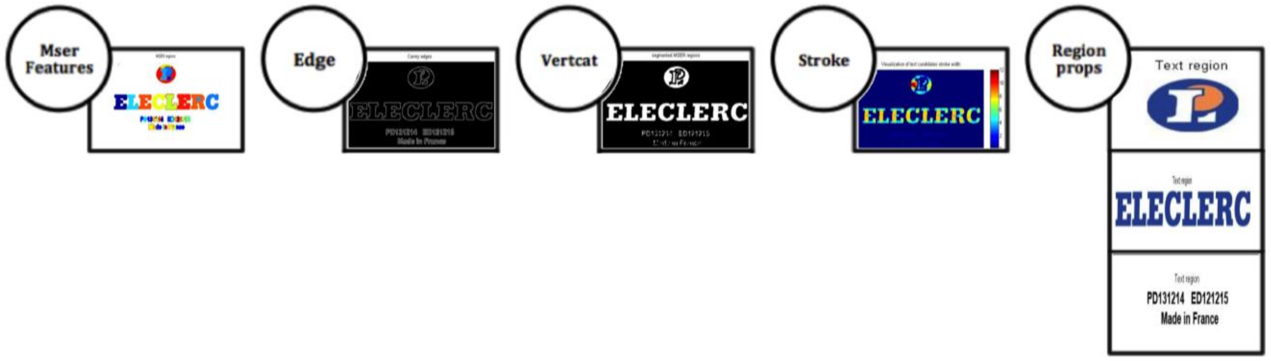


Fig. 12. Results of each operator application

The examples treated here validate practically our approach for images in quality control domain. This approach constitutes a new general and expressway of reasoning for any vision task that requires the right choice of operators and the right adjustment of their parameters, since it is based on easy mathematic operations.

C. Adjustment of DPSO parameters and discussion

Adjusting DPSO parameters is a very important step since they can have a large impact on optimization performance. In order to set proper parameters for the proposed DPSO approach some experiences will be carry on to adjust these parameters in order to see their influence on results quality [30]. Based on the examples bellow, the particle’s number and inertia weight influence on the error value has been studied, by varying them separately. The evolutions are shown in figures bellow.

- Number of particles

Selecting the proper number of particles is a critical step because it affects the algorithm performance. Number of particles needs to be sufficient to explore all possible states with the least possible iterations. In most cases, increasing the number of particles decreases the number of required algorithm iterations [31]. So the performance of the DPSO algorithm is tested with 5, 10… up to 100 particles. The number of particles effect is shown in ‘‘fig. 13’’. It is observed that the DPSO model offers the best results when the number of particles is bigger than 30 in case study 1, while in case study 2 DPSO model offers the best results when the number of particles is 30, 40, 50, 70 and 80.

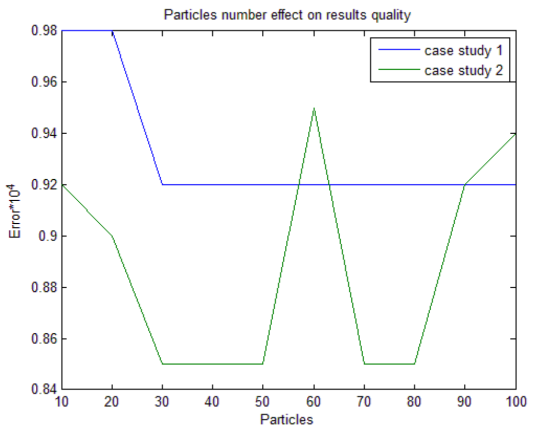


Fig. 13. Effect of particles number on performance of DPSO model

- Inertia weight

This term serves as a memory of previous velocities; it was first introduced by Shi and Eberhart [32]. To control the impact of the previous velocity: a large inertia weight favors exploration, while a small inertia weight favors exploitation [33]. The effect of inertia weight variation is shown in ‘‘Fig. 14’’. It is observed that the DPSO model offers the best result when the inertia weight is bigger than 0.5 in case study 1, while it starts from 2 in case study 2.

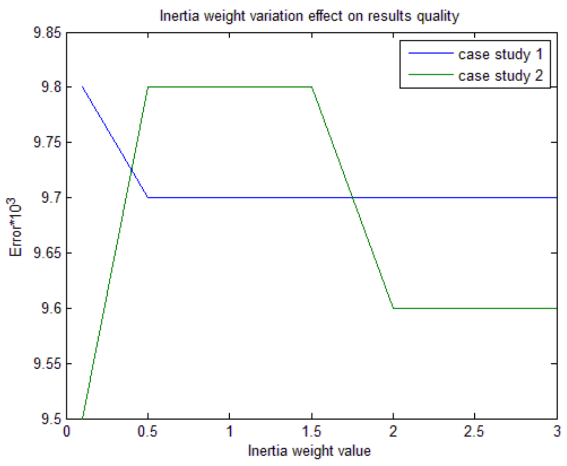


Fig. 14. Effect of inertia weight on performance of DPSO model

- Selected parameters

In regards to the parameters of DPSO, Shi and Eberhart [33], Jordehi and Jasni [34] and Pedersen [35][36] studied parameters selection on particular problems. On the basis of the above parameter analysis results and literature, Table 2 provides the detailed setting for the proposed DPSO model. It may be noted here that, the error evolution is not stationary according to figures above. We can simply locate the minimal error interval and choose the corresponding values. Effectively, there is a slight error decrease regarding the values used in Sec. 5.1.2 and Sec 5.2 (first trial) and a slight modification of the optimal values of operators' parameters but not too significant.

TABLE IV. SETTING OF THE PROPOSED DPSO MODEL

	Values for case study 1	**Values for case study 2**
Particles number	30	30
Inertia weight	0.5	2.5

Tables 5 and 6 resumes new best operators' combinations and their optimal parameters after DPSO tuning.

TABLE V. BEST OPERATORS COMBINATION AND THEIR PARAMETERS' VALUES FOR CASE STUDY 1 AFTER DPSO TUNING

		Pre-processing	**Processing**		**Post-processing**		**Error rate**
Image 1	**Operator**	Medfilt2	Edge		Bewareaopen		0.0022
	Parameters	1	Sobel	2	30	8	
Image 2	**Operator**	Medfilt2	Edge		Bewareaopen		0.0040
	Parameters	3	Prewitt	3	15	8	
Image 3	**Operator**	Ordfilt2	Edge		Bewareaopen		0.0012
	Parameters	2	Sobel	0.8	15	8	

TABLE VI. BEST PARAMETERS' VALUES FOR CASE STUDY 2 AFTER DPSO TUNING

Parameter	**Optimal Value**
Area-Range	[10, 6000]
Filter	Canny
Threshold	0,06
Eccentricity	0,995
Solidity	0,1
Width-variation	0,9
Area-threshold	10

To go further, we use a dataset of mechanical images and try to detect and recognize text on these images, using the same experimental setup described above. ‘‘Fig.15’’ shows the error rates obtained by the proposed DPSO approach with contrast to some other techniques presented in [37] [38] and [39] .

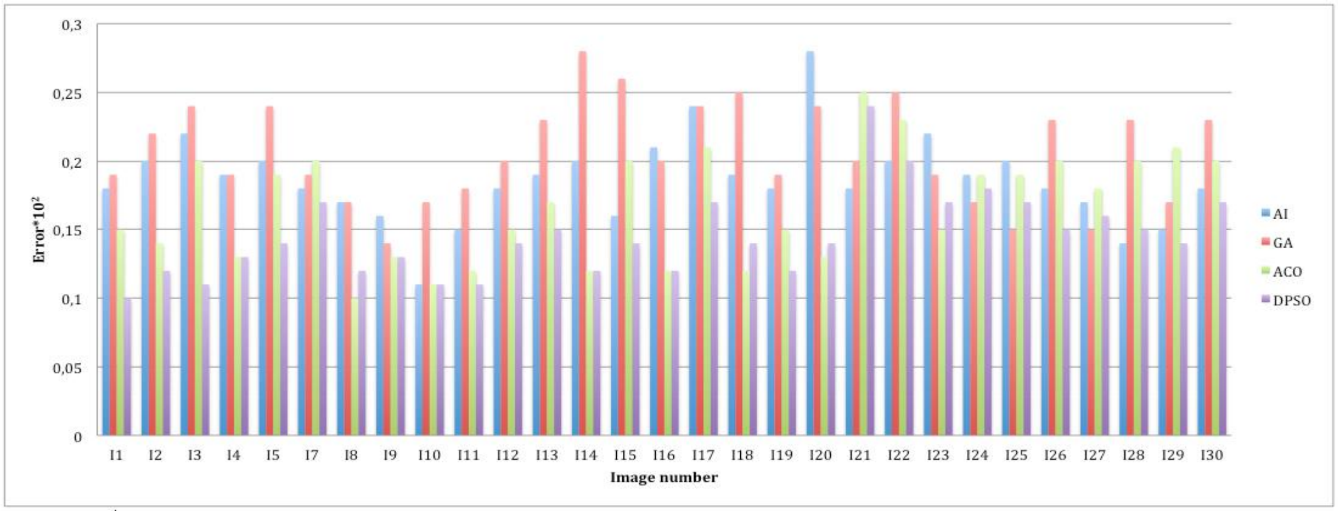


Fig. 15. Bar plot of test error with different methods. DPSO stands for Discrete Particle Swarm intelligence method proposed in this research, ACO denotes Ant Colony Optimization, GA is Genetic Algorithm, AI means artificial intelligence

Experimental results of optimal parameters specifically error rate, obtained by the proposed DPSO algorithm is compared with the ant colony optimization approach, genetic algorithm approach and artificial intelligence approach. The proposed DPSO approach obtains best results on the majority of test images, while ACO approach gives very close error values to first approach.

VII. Conclusion

Choosing the appropriate operators to apply and then adjusting their parameters values to accomplish a vision task is a very big challenge for users, in this work an automated method to optimize parameters values of image processing algorithms in quality control is presented, this system procceds automatically to decide which operator is the most appropriate to use, and adjust automatically values of its free parameters. Novel update functions for the new DPSO definition are suggested. This new system is intended to have a better precision.

In practice a set of parameters is supplied, to which a range of values is provided, with the help of DPSO the approach is applied on a specimen of test, which demonstrates the performance of the proposed procedure. A comparaison to others methods would be held to validate and support the satisfaying results of DPSO.

References

[1] B. Nickolay, B. Schneider, S.Jacob, "Parameter Optimization of an Image Processing System using Evolutionary Algorithms", *Proc. of the 7th International Conf on Computer Analysis of Images and Patterns*, Germany, 10–12, 1997.

[2] G.W.Taylor, "A Reinforcement Learning Framework for Parameter Control in Computer Vision Applications" *IEEE Proc. of the First Canadian Conf. on Computer and Robot Vision*, 2004.

[3] Y. H. Du, J. Fang, and C. Miao, "Frequency-domain system identification of an unmanned helicopter based on an adaptive genetic algorithm," *IEEE Trans. Ind. Electron*., vol. 61, pp. 870–881, 2014.

[4] M.Dorigo and M.Birattari, "Ant Colony Optimization", *Encyclopedia of Machine Learning*, pp 36-39, 2010.

[5] Y.Bao, Z.Hu, T.Xiong, "A PSO and pattern search based memetic algorithm for SVMs parameters optimization", Neurocomputing, pp 98–106, 2013.

[6] L.Benchikhi, M.Sadgal, A.El fazziki, "Optimization of parameters values in industrial vision algorithms", *Proc. of the International Symposium on Operational Research and Applications*, p 145-146, Marrakech, 2013.

[7] S.Treuillet, D. Driouchi and P. Ribereau, "Adjustment of parameters in an image processing chain by an experimental 2k-p factorial designs", *traitement du signal*, volume 21, numéro 2, 2004.

[8] L.Franek and X.Jiang, "Orthogonal design of experiments for parameter learning in image segmentation Signal Processing*", Signal Processing,* Volume 93, Issue 6, Pages 1694–1704, 2013.

[9] A.J. Pretorius, D. Magee, D. Treanor and R.A. Ruddle, "Visual Parameter Optimization for Biomedical Image Analysis: A Case Study", *Proc. of SIGRAD*, 2012.

[10] L.Vila, "A survey on temporal reasoning in artificial intelligence", *Ai Communications,* 7.1: 4-28, 1994.

[11] I.Qaffou, S.Mohammed, A.El fazziki, "A Multi-Agents Architecture to Learn Vision Operators and their Parameters", *International Journal of Computer Science Issues* , p140, 2012.

[12] S.Szenasi, "Genetic algorithm for parameter optimization of image segmentation algorithm", *IEEE Proc. Of the International Symposium on Computational Intelligence and Informatics*, 2013.

[13] L.Benchikhi, M.Sadgal and A.Elfazziki, "An Optimization approach of parameters in image processing based on PSO: case of quality control", Proc. Of the International Symposium on Operational Research ans applications, pp 145-146, Marrakech, 2013.

[14] J. Kennedy and R. Eberhart, "Particle Swarm Optimization". *IEEE Proc. of International Conf. on Neural Networks*, vol. 6, pp. 1942–1948, Piscataway, December 1995.

[15] A.Chen, G.Yang and Z.Wu, "Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem", *Journal of Zhejiang University Science*, 7(4):607–614, 2006.

[16] W.Pang, P.K.Wang, G.Zhou and L.Dong, "Fuzzy discrete particle swarm optimization for solving traveling salesman problem", *Proc. of the Fourth International Conf. on Computer and Information Technology*, 796–800, 2004.

[17] D.Y.Shaa, H.Lin, "A multi-objective PSO for job-shop scheduling problems", *Expert Systems with Applications*, pp 1065-1070, 2010.

[18] D.Anghinolfi, M.Paolucci, "A new discrete particle swarm optimization approach for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times", European Journal of Operational Research, 193(1):73–85, 2009.

[19] C.Wong, H.Wang, and S.Li, "PSO-based Motion Fuzzy Controller Design for Mobile Robots", International Journal of Fuzzy System, Volume 10, 2008.

[20] V.Ranaee, A.Ebrahimzadeh, R.Ghaderi, "Application of the PSO–SVM model for recognition of control chart patterns, *ISA Transactions*, volum 49, 2010.

[21] N.Arana-Daniel, A.A.Gallegos, C.Lopez-Franc and A.YAlanis, "Smooth global and local path planning for mobile robot using particle swarm optimization, radial basis functions, splines and Bézier curves". *IEEE Congress on Evolutionary Computation*, 2014.

[22] F.Mohsen, M.Hadhoud, K.Mostafa and K.Amin, "A New Image Segmentation Method Based on Particle Swarm Optimization", *The International Arab Journal of Information Technology*, 2012.

[23] R.Yadav, D.Mandal. "Speaker Recognition using Particle Swarm Optimization", *International Journal of Electronics & Communication Technology*, 2011.

[24] L.Wen-sheng,Y.Qiong, D.Chun-jian, "Application of the BP Neural Network Based on PSO in Dynamic Gesture Recognition", Computer Engineering & Science, 2011.

[25] J.Mohammed, M.Basalamah, A.Majid and A.Sid-Ahmed, "Computer Vision-Based Quality Inspection System of Transparent Gelatin Capsules in Pharmaceutical Applications*", American Journal of Intelligent Systems*, 2(1): 14-22, 2012.

[26] J.Kennedy and R.Eberhart, "A discrete binary version of the particle swarm algorithm", *Proc. of the IEEE International Conf. on Systems, Man, and Cybernetics*, pp. 4104 - 4109, Florida, 1997.

[27] G.Palermo, C.Silvano and V.Zaccaria, "Discrete Particle Swarm Optimization for Multi-objective Design Space Exploration", *IEEE Proc. Of the 11th EUROMICRO Conf. on Digital System Design Architectures*, Methods and Tools, 2008.

[28] M.Khanesar, H.Tavakoli, M.Teshnehlab and M.Shoorehdeli, "Novel Binary Particle Swarm Optimization", *Particle Swarm Optimization*, 2009.

[29] M.Clerc, "Discrete particle swarm optimization, illustrated by the traveling salesman problem. In: Studies in Fuzziness and Soft Computing New optimization techniques in engineering", *Babu, B.V. & Onwubolu, G.C,* pp. 219–239, Springer ISBN: 978-3-5402-0167-0, Berlin, 2004.

[30] A.Jordehi & J. Jasni, "Parameter selection in particle swarm optimisation: a survey", *Journal of Experimental & Theoretical Artificial Intelligence*, 25:4, 527-542, 2013.

[31] Y.Shi and R.Eberhart, "Comparing inertia weights and constriction factors in particle swarm optimization", *Proc. of the Congress on Evolutionary Computation 1*, pp. 84–88, 2000.

[32] Y.Shi and R.Eberhart, "Parameter selection in particle swarm optimization", *Proc. of Evolutionary Programming VII*, pp. 591–600, 1998.

[33] Y.Shi and R.Eberhart, “Parameter selection in particle swarm optimization", *Evolutionary programming VII*. Springer Berlin Heidelberg, 1998.

[34] A.Jordehi and J.Jasni, "Parameter selection in particle swarm optimisation: a survey", *Journal of Experimental & Theoretical Artificial Intelligence*, 25.4: 527-542, 2013.

[35] Pedersen, “*Tuning & Simplifying Heuristical Optimization*”, PHD THESIS, University of Southampton, School of Engineering Sciences, Computational Engineering and Design Group, 2010.

[36] Pedersen, “Good parameters for particle swarm optimization” , Technical Report (Hvass Laboratories), 2010.

[37] I.Qaffou, M.Sadgal, A.Elfazziki, ‘‘A New Automatic Method to Adjust Parameters for Object Recognition’’, IJACSA, 2012.

[38] L.Benchikhi, M.Sadgal, A.Elfazziki and F.Mansouri, ‘‘An ant colony based model to optimize parameters in industrial vision’’, IJAI ‘‘in press’’, 2015.

[39] S.Bolme, J.Beveridge, A.Draper, P.Phillips, and Y.Lui, ‘‘Automatically Searching for Optimal Parameter Settings Using a Genetic Algorithm’’, Proceedings of the 8th International Conference, ICVS 2011, Sophia Antipolis, France, September 20-22, 2011.