T.C.

ALTINBAŞ UNIVERSITY

Graduate School of Science and Engineering

Electrical and Computer Engineering

ROBOT PATH PLANNING BY AREA SEGMENTATION USING THE FUZZY C-MEANS ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Abdulrahman Tareq Ali AL-KHAYYAT

Master Thesis

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2021

ROBOT PATH PLANNING BY AREA SEGMENTATION USING THE FUZZY C-MEANS ALGORITHM AND PARTICLE SWARM OPTIMIZATION

by

Abdulrahman Tareq Ali AL-KHAYYAT

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master Thesis

ALTINBAŞ UNIVERSITY

2021

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of master of science.

____

Asst. Prof. Dr. Abdullahi IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of engineering and natural science,	
	Altinbas University	________________
Asst. Prof. Dr. Çağatay AYDIN	School of engineering and natural science,,	
	Altinbas University	________________
Asst. Prof. Dr. Izzet Parug DURU	Vocational School,	
	Istanbul Gedik Univeristy	________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies:

____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Abdulrahman Tareq Ali AL-KHAYYAT

DEDICATION

I would like to thank Allah Almighty for my success in my studies.

My dear father and my first teacher and my role model. I know very well that you loved science and study,

That is why you have always urged us to strive hard in studying and excel in order to obtain the highest degrees and obtain masters and doctorates.

But I now know that obtaining a master's degree will make you very happy and proud of me.

And also dedicate This thesis is to dear my father and mother .as there are no words to describe what they mean to me, and there is nothing I can do for what I have given for me.

And I will continue to do my best to achieve your expections.

Lastly, I dedication this to my, brothers and sisters (Raniah, Mohammed, Abdullah, Esraa, Abdulmohaimen and Abdulsallam) who have been encouraging and support me during this study.

ACKNOWLEDGEMENTS

I would like to thank my supervisor professor Ibrahim for all support during my study. It’s great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

ABSTRACT

ROBOT PATH PLANNING BY AREA SEGMENTATION USING THE FUZZY C-MEANS ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Abdulrahman Tareq Ali AL-KHAYYAT

M.Sc., Electrical and Computer Engineering, Altınbaş University

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 03/2021

Pages: 55

Mobile robotics is a valuable tool for exploring environments inaccessible to humans due to their remoteness, cost or danger, and for performing unpleasant or laborious tasks. It is a relatively new field, until recently experimental, but it is already being applied to real problems with satisfactory result. In this work we have designed and implemented a system whose main purpose is to plan trajectories for this mobile robot. The main goal of this work is to decrease the cost of hardware implementation by using 3d design to simulate and environment which the robot can find the optimal path in it, we use the fuzzy C-means algorithm to cluster the image of the robot camera and the gray wolf optimization algorithm to decide which path to take.

Keywords: Optimization, energy, Fuzzy -C means, Grey wolf, Nodes, Robot.

ÖZET

ROBOT PATHPLANNING BY AREA SEGMENTATION USING THE FUZZY C-MEANS ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Abdulrahman Tareq Ali AL-KHAYYAT

M.Sc., Electrical and Computer Engineering, Altınbaş University

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 03/2021

Pages: 55

Mobil robotik, uzaklıkları, maliyetleri veya tehlikeleri nedeniyle insanların erişemeyeceği ortamları keşfetmek ve hoş olmayan veya zahmetli görevleri yerine getirmek için değerli bir araçtır. Yakın zamana kadar deneysel olan nispeten yeni bir alandır, ancak tatmin edici sonuçları olan gerçek problemlere zaten uygulanmaktadır. Bu çalışmada, ana amacı bu mobil robot için yörüngeler planlamak olan bir sistem tasarladık ve uyguladık. Bu çalışmanın temel amacı, robotun içinde en uygun yolu bulabileceği simülasyonu ve ortamı 3d tasarımı kullanarak donanım uygulama maliyetini düşürmektir, robot kameranın görüntüsünü kümelemek için bulanık C-araçlar algoritmasını kullanıyoruz ve hangi yolun izleneceğine karar veren gri kurt optimizasyon algoritması

Anahtar Kelimeler: Optimizasyon, enerji, Bulanık -C anlamı, Bozkurt, Düğümler, Robot.

TABLE OF CONTENTS

Pages

DEDICATION……………………………………………………………………………………v

ACKNOWLEDGEMENTS……………………………………………………………….…….vi

ABSTRACT... vii

LIST OF TABLES x

LIST OF FIGURES xii

LIST OF ABBREVIATIONS xiii

LIST OF SYMBOLS..xiv

1. INTRODUCTION..15

1.1 BACKGROUND 15

1.2 MOTIVATION 15

1.3 PROBLEM STATEMENT...15

1.4 CONTRIBUTION..16

1.5 THESIS ORGANIZATION 16

2. LITRITAURE REVIEW 17

3. MATERIALS AND METHODS 18

3.1 ROBOT PATH PLANNING 18

3.2 ARTIFICIAL VISION 22

3.3 FUZZY C-MEANS (FCM) 25

3.4 PARTICLE SWARM OPTIMIZATION 30

3.5 KNN CLASSIFICATION 37

4. PROPOSED METHOD **..43**

5. CONCLUSION **..48**

REFERENCES **..49**

LIST OF TABLES

Pages

Table 5.1: MSE and PSNR of the FCM algorithm compared to methods of segmentation…………..…...46

LIST OF FIGURES

Pages

Figure 1.1: Robot path planning 15

Figure 1.2: Path planning by area segmentation..16

Figure 3.1 Simple robot path planning scheme 18

Figure 3.2 Honda the E0 prototype 20

Figure 3.3 1999 Sony launched AIBO 21

Figure 3.4 Artificial vision in robotics 23

Figure 3.5 Applications of Artificial Vision..24

Figure 3.6 Fuzzy C-means diagram 26

Figure 3.7 The FCM algorithm..27

Figure 3.8 FCM minimization 28

Figure 3.9 FCM Clusters..29

Figure 3.10 Applications of swarm algorithms. 30

Figure 3.11 Types of Swarm intelligence SI algorithm 32

Figure 3.12 Diagram of PSO workflow 33

Figure 3.13 description of the behavior of PSO iteration 34

Figure 3.14 Parameter space of PSO...35

Figure 3.15 PSO vs UPS 36

Figure 3.16 KNN classification 37

Figure 3.17 KNN voting...38

Figure 3.18 KNN catagories 40

Figure 3.19 KNN neighbors 41

Figure 3.20 KNN Accuracy..41

Figure 4.1 Environment set up in Autodesk Maya 43

Figure 4.2 Front view camera image..44

Figure 4.3 FCM segmentation of the image 44

Figure 4.4 Proposed robot path planning simulation 45

Figure 4.5 The change in X, Y, Z position of the mobile robot 45

FFigure 4.6 KNN accuracy and time compared to other methods 46

Figure 4.7 PSO accuracy and time compared to the Genetic algorithm 47

LIST OF ABBREVIATIONS

KNN	:	K nearest number
DWT	:	Discrete Wavelet Transform
CNN	:	Convolutional neural network
PCA	:	Principal Component Analysis
NN	:	Neural network
GD	:	Gender Detection

LIST OF SYMBOLS

μ : Mobility of electron

λ : Wave length

ω : Angle frequency

1. INTRODUCTION

1.1 BACKGROUND

Mobile robotics is a valuable tool for exploring environments inaccessible to humans due to their remoteness, cost or danger, and for performing unpleasant or laborious tasks. It is a relatively new field, until recently experimental, but it is already being applied to real problems with satisfactory results:

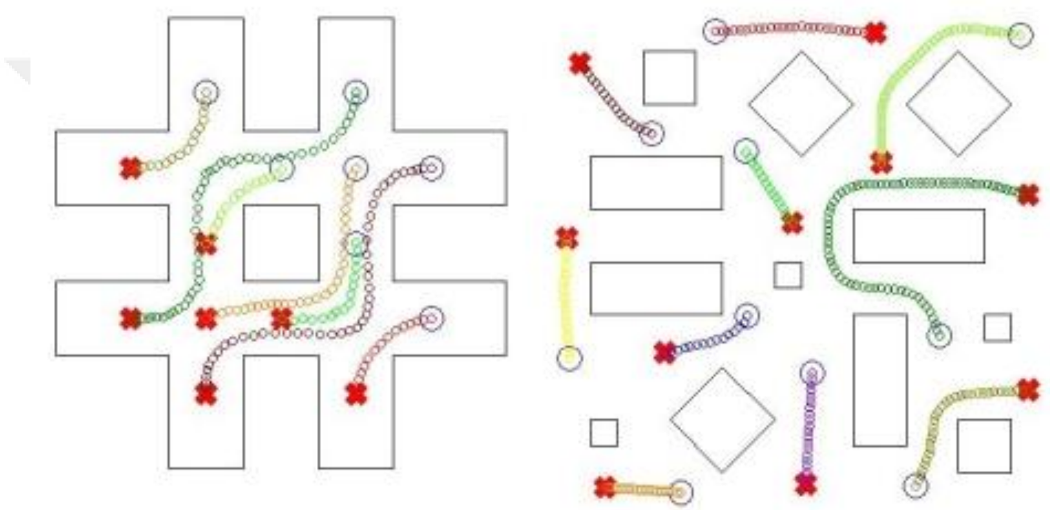


Figure 1.1: Robot path planning.

1.2 MOTIVATION

Greater autonomy of the robot, that is, less human supervision for its correct operation is one of the primary purposes of this area of knowledge. Achieving that goal poses numerous challenges. The most obvious is the locomotion itself that must be complemented with the perception of the environment, the positioning in it, the calculation of trajectories and their navigation.

1.3 PROBLEM STATEMENT

3D simulation plays a relevant role in both the design and control of mobile robots. It allows to reflect the situation of the real robot or simulate hypothetical scenarios. At the same time, it provides a graphical interface for robot control:

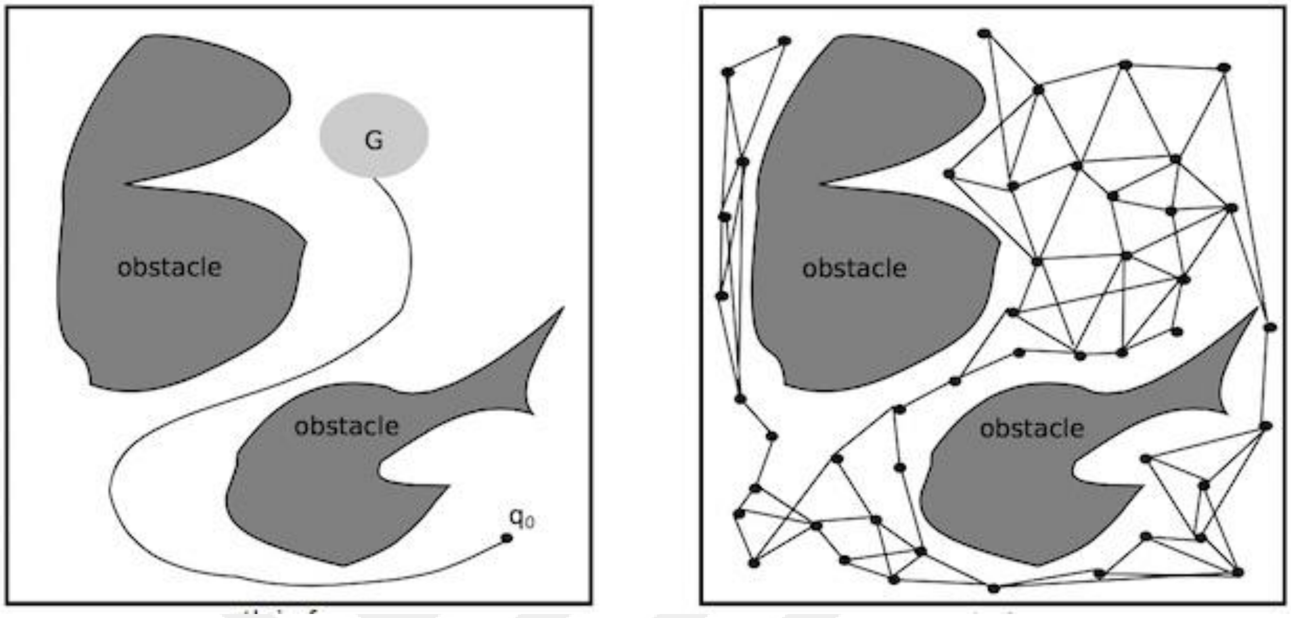


Figure 1.2: Path planning by area segmentation.

1.4 CONTRIBUTION

In this work we have designed and implemented a system whose main purpose is to plan trajectories for this mobile robot. The main goal of this work is to decrease the cost of hardware implementation by using 3d design to simulate and environment which the robot can find the optimal path in it, we use the fuzzy C-means algorithm to cluster the image of the robot camera and the gray wolf optimization algorithm to decide which path to take.

1.5 THESIS ORGANIZATION

The thesis is structured as follows: Section 2 is where we review some of the previous work and implementation on smart grids. Section 3 is where we give a background about all the components. Section 4 is where we explain our model in details and implementation of that model in details, Section 5 is where we simulate our model and record the results obtained. Section 6 is where we conclude our work and put a future scope into perspective.

2. LITERATURE REVIEW

Path planning involves guiding the robot from a start point to an end point, through an obstacle-free route. A movement planning algorithm must produce, from the objective to be achieved and the available data from the environment in which the navigation is to be carried out, a series of commands that allow the robot to travel safely in an environment with obstacles until completing the task. In order to address the problem, it is necessary to store information about the robot's environment at all times. It is important that this information is as up-to-date as possible, since based on it the planning algorithm makes decisions about which is the best route among all the possible ones that connects the current state and the target, Wang et.al [1] proposed a path planning scheme for manufacturing robots with artificial vision based on the particle swarm optimization, while Shao et.al [2] proposed a learning transfer scheme derived from the swarming behavior of the drone swarm, and applied it to robotics, Ravankar et.al [3] proposed an obstacle free path planning algorithm in multiple cooperative robots by also using the transfer learning scheme, Tao et.al [7] implemented a scheme for path planning in mobile robots using the dynamic inver-over evolutionary algorithm, which takes tame as an end condition for the path calculations, Ma et.al [8] proposed a scheme for path planning in coal mining robots using a hybrid Dijkstra algorithm and ant colony optimization, Miao et.al [9] proposed a coverage path planning in cleaning robots based on the rectangular map decomposition and proved to have decreased the coverage time down to 14 times compared with regular methods, Nakamura et.al [10] proposed a scheme for robot path planning in large areas with motion disturbance by decreasing the number of cells that the robot must pass through, in this paper, we propose a fast scheme for robot path planning by using the FCM segmentation of the images taken from the front view of the robot camera, and the fast k-nn classification due to the fact that k-nn does not need any training time therefore the time needed for decision making is reduced substantially, the rest of the paper is organized as follows: we will be explaining the methods and methodology in section 2, while in section 3 we will explain our proposed method, in section 4 we will put our results and in section 5 we will conclude our work.

3. MATERIALS AND METHODS

In this chapter we review some of the essential components that we used to implement our model, we give a brief introduction to the component, and then we show how we simulate that component inside MATLAB environment.

3.1 ROBOT PATH PLANNING

Movement planning involves guiding the robot from a start point to an end point, through an obstacle-free route. A movement planning algorithm must produce, from the objective to be achieved and the available data from the environment in which the navigation is to be carried out, a series of commands that allow the robot to travel safely in an environment with obstacles until completing the task. In order to address the problem, it is necessary to store information about the robot's environment at all times. Best suited for a 2-D motion planning problem is a grid map. It is important that this information is as up-to-date as possible, since based on it the planning algorithm makes decisions about which is the best route among all the possible ones that connects the current state and the target.

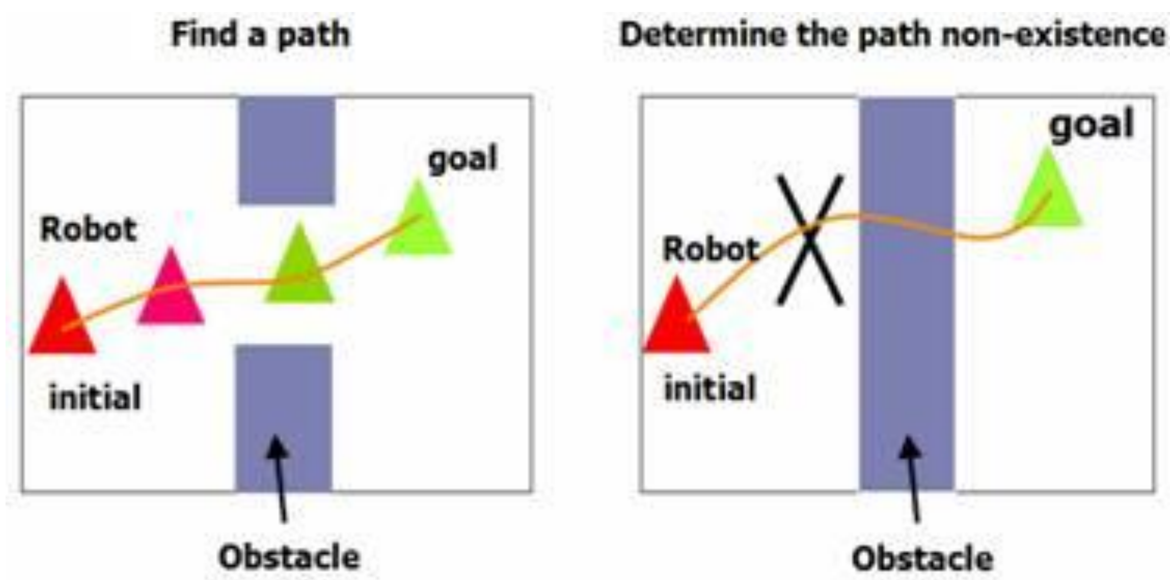


Figure 3.1 Simple robot path planning scheme

Having a priori information on the environment map helps the initial solution provided by the planner more closely to the problem posed, and therefore reduces the number of re-planning to be carried out during execution. Both in case of having a previous map or otherwise, the system must take the information from the sensors with which the robot is equipped at all times, and combine the old information with the new one. For this reason, the solution provided cannot be static, as it can be valid at that time, but not if there are changes in the environment that affect the planned path. In this case, according to the changes that have occurred, it is necessary to check to what extent they affect the previous solution, and refine it until it is adequate to the new arrangement of obstacles in the environment of the robot. path planning has been proposed as a directed graph, where the nodes are the robot states, defined in five dimensions: $(x, y, \theta, v, \omega)$, and the arcs are the discrete movements, or primitive paths, which are the control actions that can be carried out by the robot that allow traffic between them. Each of these movements is associated with a specific control order, so the solution provided by the search algorithm is a composition of these movements, and therefore control orders, that allow reaching the target point based on the current state. of the robot. Both the states of the robot and the set of actions are discrete, so planning has been proposed as a solvable process using a search algorithm. The first mobile robots appear in the middle of the 20th century. William Gray Walter built two autonomous mobile robots, Elmer and Elsie, in 1949. Between 1961 and 1963, Beast was developed at Johns Hopkins University. Beast is capable of identifying and approaching electrical outlets to recharge your battery autonomously. In the late 1960s, Shakey was developed at the Stanford Research Institute. Shakey is the first mobile robot capable of reasoning about its own actions. While other robots had to be directed by simple actions to perform a complex task, Shakey analyzed the complex task and reduced it to a sequence of simple actions. From this moment on, the research, design and construction of mobile robots grew exponentially:

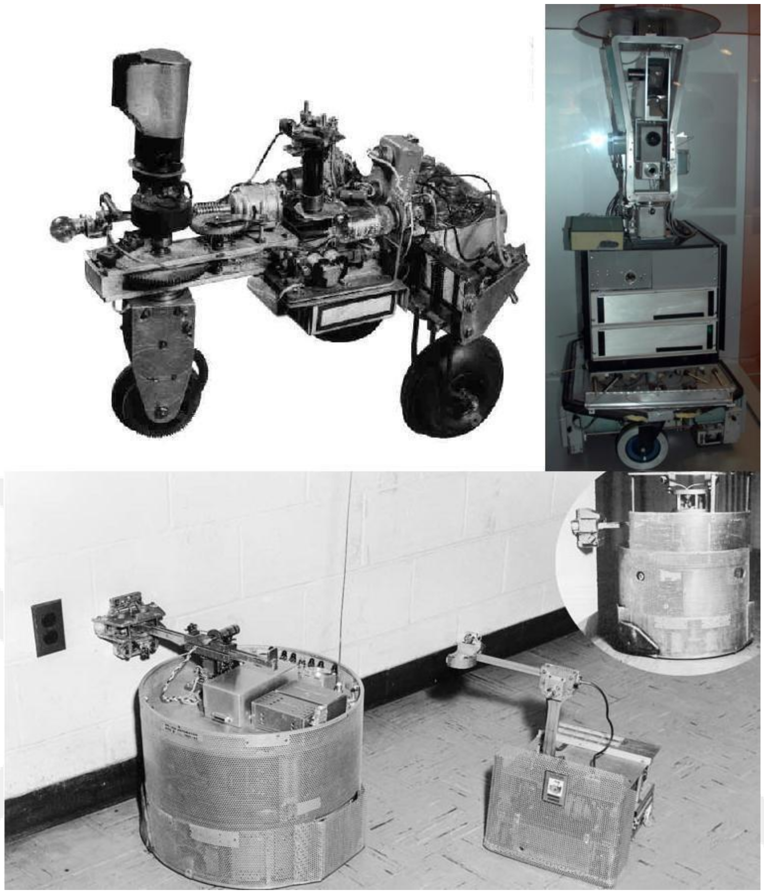


Figure 3.2 Honda the E0 prototype

In 1986 the Japanese company Honda created the E0 prototype. It is a biped robot capable of walking slowly in a straight line. It took about 5 seconds to take a step. The E0 was followed by new generations of prototypes that were introducing new improvements. In 1993 the prototype E6 walked autonomously at human speed and circumvented simple obstacles, but lacked trunk and upper limbs. These elements would be incorporated in the next generation, in which prototypes P1, P2 and P3 were developed. In 2000 Honda presented the first version of ASIMO. ASIMO was lighter and more flexible than prototypes, improved movement, and was friendlier in appearance. Five years later the second version appeared. This model, among other improvements, is capable of running

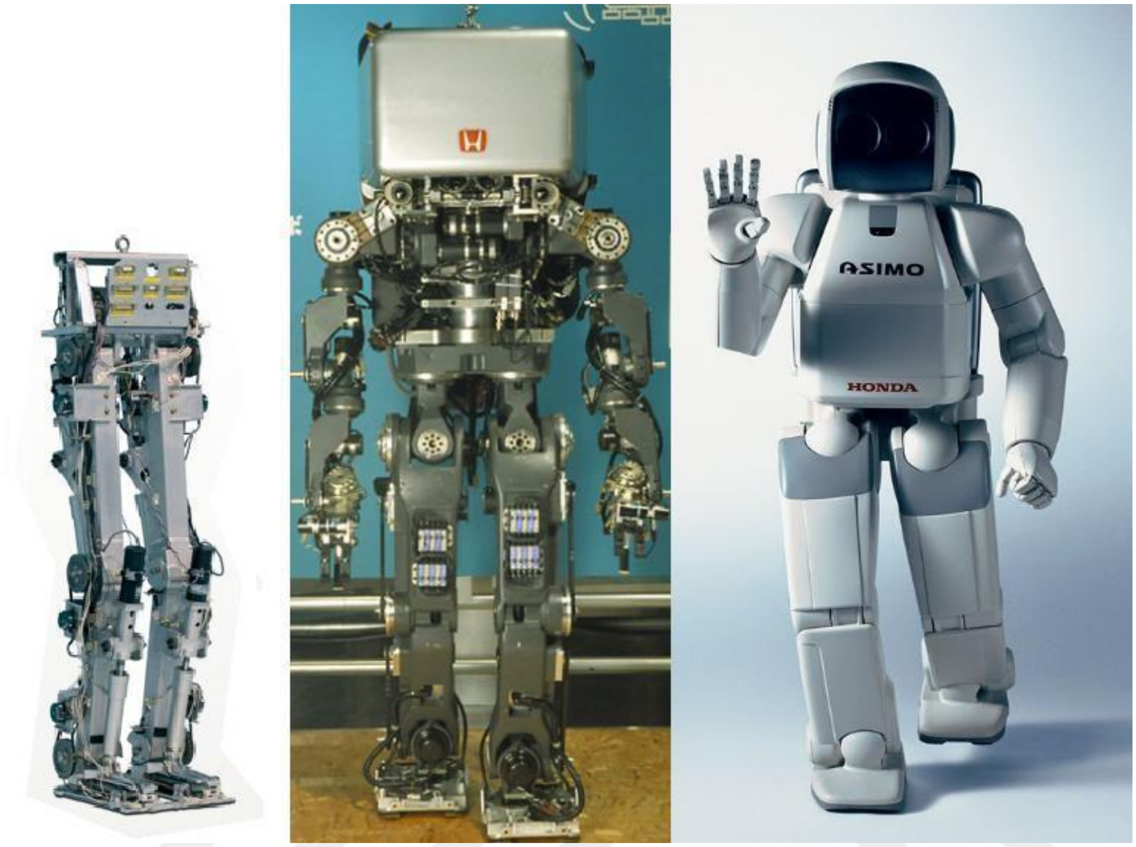


Figure 3.3 1999 Sony launched AIBO

Mobile robots have been successfully used for practical purposes, for example in the exploration of other planets. On July 4, 1997 the Sojourner rover became the first mobile robot to reach the surface of Mars. It was controlled by remote control from Earth, although it included autonomous navigation using a laser to detect the presence of obstacles. In 2004 twin rovers Spirit and Opportunity arrived on the Red Planet on the Mars Exploration Rover mission. They belong to a more technologically advanced model than the Sojourner. Navigation has been qualitatively improved in this new model thanks to the incorporation of a stereoscopic vision system for the recognition of obstacles. Both rovers are still operating on the surface of Mars today. In 1999 Sony launched AIBO. It is a pet robot with the appearance of a dog. Equipped with a large number of sensors, AIBO is capable of perceiving external stimuli from the environment or its owner and acting accordingly. Highlights your learning ability.

In recent years, robots designed to carry out household tasks autonomously have started to be commercialized. Among them we can name iRobot Roomba cleaning robot and Husqvarna Auto mower lawn mower.

3.2 ARTIFICIAL VISION

One of the purposes of machine vision is to make a computer or robot "understand" a scene or the characteristics of an image and that this understanding is useful to carry out a mission. Stereoscopic vision is one of the artificial vision techniques where there is more than one view of a scene, taken from different points. Through these images the three-dimensional characteristics of the scene are extracted and the study relevant to the application in this case is made thanks to obtaining a disparity map and subsequent treatment based on colors. Naturally, our vision mechanism is stereo, that is, we are able to appreciate, through binocular vision, the different distances and volumes in the environment that surrounds us. Our eyes, due to their separation, obtain two images with small differences between them, which we call disparity. Our brain processes the differences between the two images and interprets them in such a way that we perceive the sensation of depth, distance or closeness of the objects that surround us. This process is called stereopsis. Various mechanisms intervene in the stereopsis. When we observe very distant objects, the optical axes of our eyes are parallel.

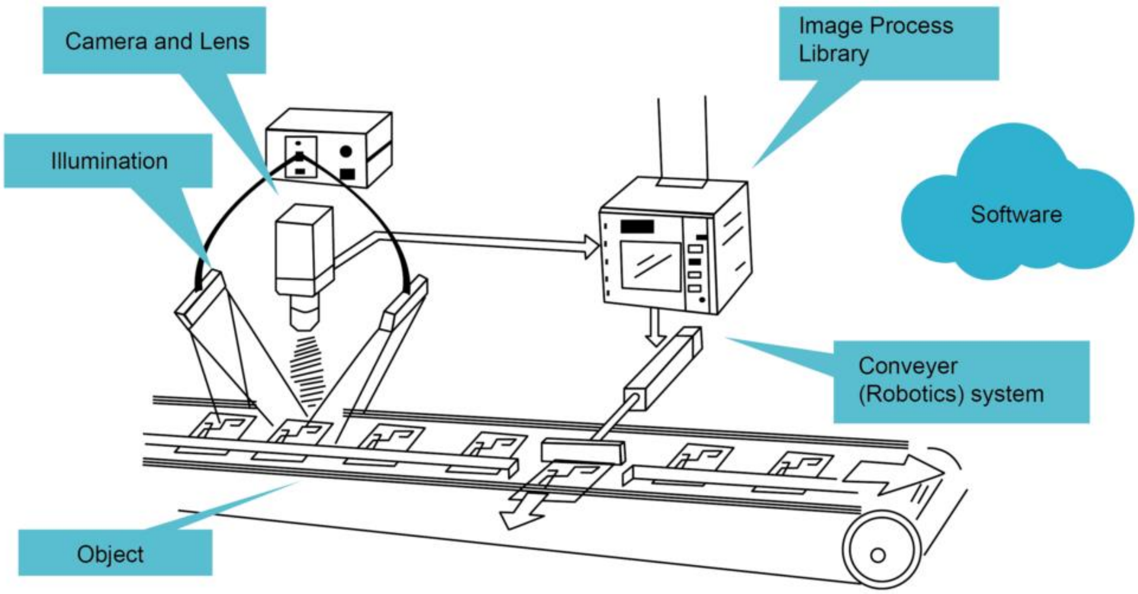


Figure 3.4 Artificial vision in robotics

When we observe a nearby object, our eyes rotate so that the optical axes are aligned on it, that is, they converge. At the same time, the accommodation or focus is produced to clearly see the object. This joint process is called a merger. One factor directly involved in this capacity is interocular separation. The greater the separation between the eyes, the greater the distance at which we can appreciate the relief effect. This applies, for example, in binoculars, where, by means of prisms, an effective interocular separation greater than normal is achieved, which means that distant objects can be seen in relief that under normal conditions we would not be able to separate from the environment. This technique is applied in aerial photography, in which stereoscopic pairs with separations of hundreds of meters are obtained and in which it is possible to clearly appreciate the relief of the terrain, which with normal vision and from a great height would be impossible. One of the oldest practical applications is the visualization and measurement of the land relief using aerial photographs. If an airplane takes two photographs of a terrain area with a certain calculated distance between them, a stereo-pair is obtained, which can later be seen in relief with a special stereoscope. If the shots are taken with adequate precision, they allow calculating elevations in the field, for which stereo comparators are used. Simulated 3D images can also be generated from terrain data using specific software. Navigation is the science of driving a mobile robot while traversing an environment to reach a destination or goal without colliding with any obstacles. Navigation leads us to create a series of tools totally necessary for creating a viable and safe path. We have to create a map with the information we have or create it as we explore. For this we have to take into account several important variables such as the perception of the environment, sensor fusion, movement control Planning is to foresee and decide today the actions that can take us from the present to a desirable future, it is not about making predictions about the future but making the pertinent decisions for that future to occur. The general idea in path planning is to decouple the problem of robot kinematics and dynamics from that of obtaining a path (optimal or not) free of obstacles. Of all the methods that can be used, the most recommended in our case for speed, types of sensors to be used, terrain topology and others is a method of occupying fixed cells. Basically, the workspace is divided into cells of a certain area.

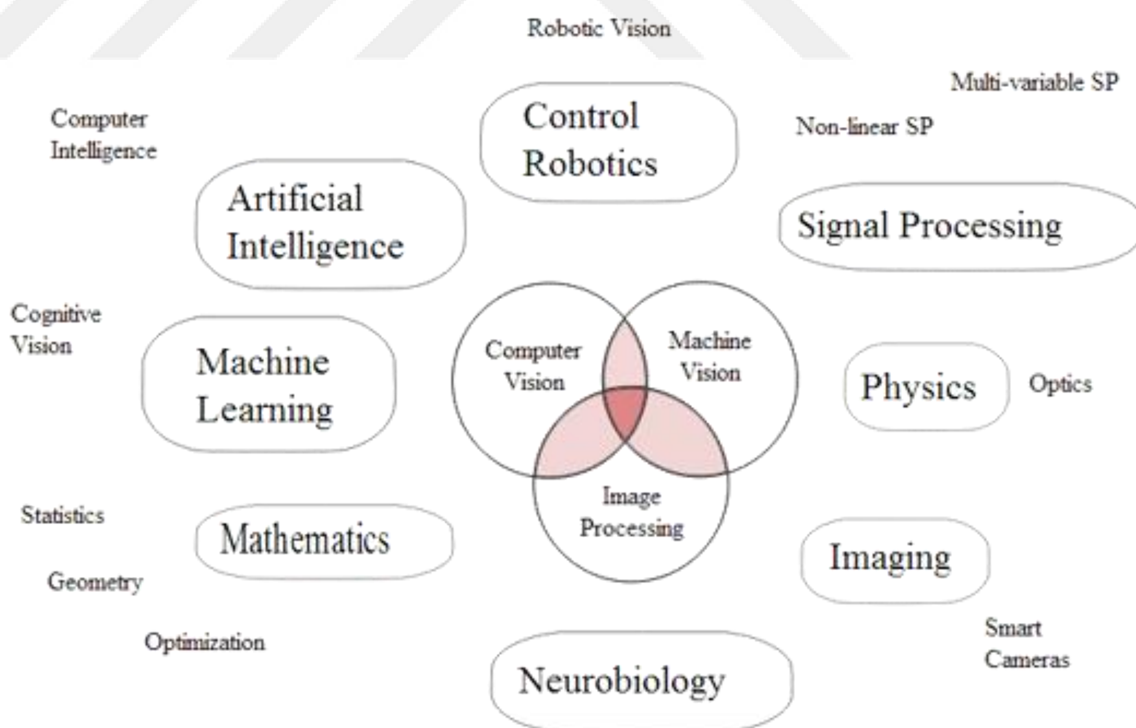


Figure 3.5 Applications of artificial vision

3.3 FUZZY C-MEANS (FCM)

The FCM algorithm is one of the most popular grouping techniques, this algorithm was developed by Dunn [4] and eventually modified by Bezdek [2] in 1981. FCM is a fuzzy and iterative algorithm that is part of the techniques of unsupervised grouping and its objective is to find patterns or groups in a certain data set, so this type of pattern is used to classify information [16]. The basic principle of operation of this algorithm is to group or separate the data provided in groups called clusters. The advantage of this algorithm to others that also work on the principle of grouping or clustering, is that the data can be close enough to two groups in such a way that it is difficult to label any of these. The FCM is an algorithm developed to solve these problems, making a partition data set smooth and restricted, that is; with these partitions in which all the data have a degree of belonging for all the groups, and the sum of these degrees of belonging is equal to 1. For the development of this investigation, 2 groups are presented: the first corresponding to the body of the whale. blue and the second contains the information from the sea [9]. Suppose that X is a data set and an element belonging to X is xi, a partition P = {C1, C2, ..., CC} where c indicates the number of clusters. P is decided to be a soft and constrained partition of X when the following conditions are met:

$$\forall x_i \in X \quad \forall C_j \in P \qquad 0 \leq \mu c_j(x_i) \leq 1$$

$$\forall x_i \in X \qquad \exists C_j \in P \;\; tal\ que \;\; \mu c_j(x_i) > 0$$

$\mu c_j(x_i)$ $\qquad$ x_i

$$\textstyle\sum_j \mu C_j(x_i) = 1 \;\; \forall x_i \in X$$

Mathematical morphology is a well-established technique for image analysis, with solid mathematical foundations [16] that find enormous applications in many areas, primarily image analysis [14]. Most practical applications are based on a combination of a small set of operations called: erosion and dilation. Mathematical morphology was developed initially for binary images and later in generalized images with gray values [15]. The fundamental instrument in mathematical morphology is the structuring element. A structuring element is simply like a pixel pattern in which it is defined from an origin [12].

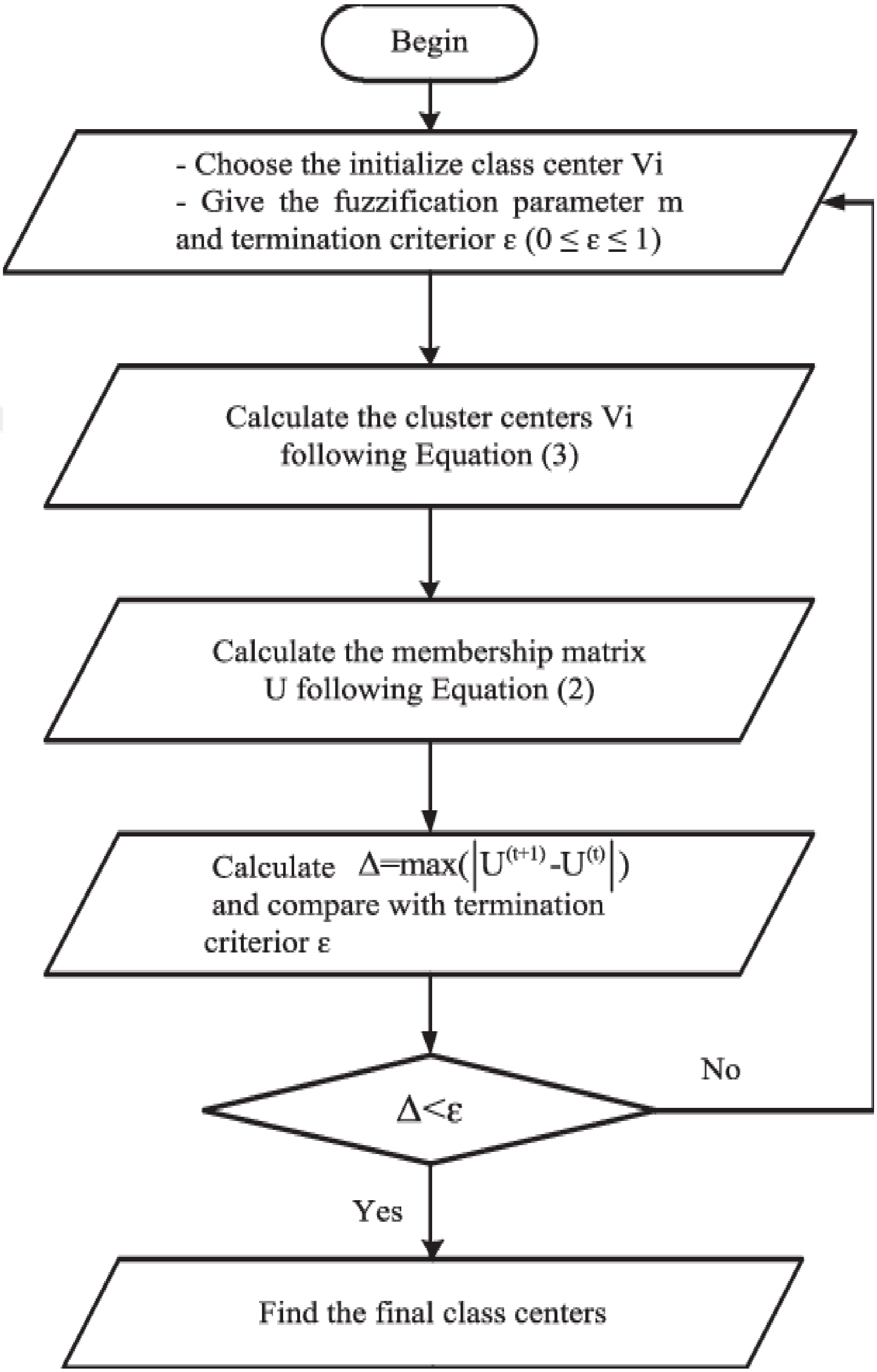


Figure 3.6 Fuzzy C-means diagram

The use of grouping methods in segmentation tasks is common. The most widely used algorithms consist of hard partitions like KMeans and soft partitions like C-Means [1,2]; In recent years, clustering of pixels in atomic regions perceptually called Cluster has become important [5-8]. A Cluster consists of a grouping of pixels in perceptually regional atomic regions; specified Cluster an appropriate way to calculate the local characteristics of an image. They capture redundancy in the image [5] and greatly reduce the complexity of subsequent image processing tasks. There are various approaches to grouping Cluster; Achanta et al. Compares overall performance across several of the most representative algorithms [8] plus makes a quick analysis of the performance of 4 representative Cluster algorithms applied to a diverse set of images [7], including magnetic resonance imaging (MRI) and electron micrographs. (EM).

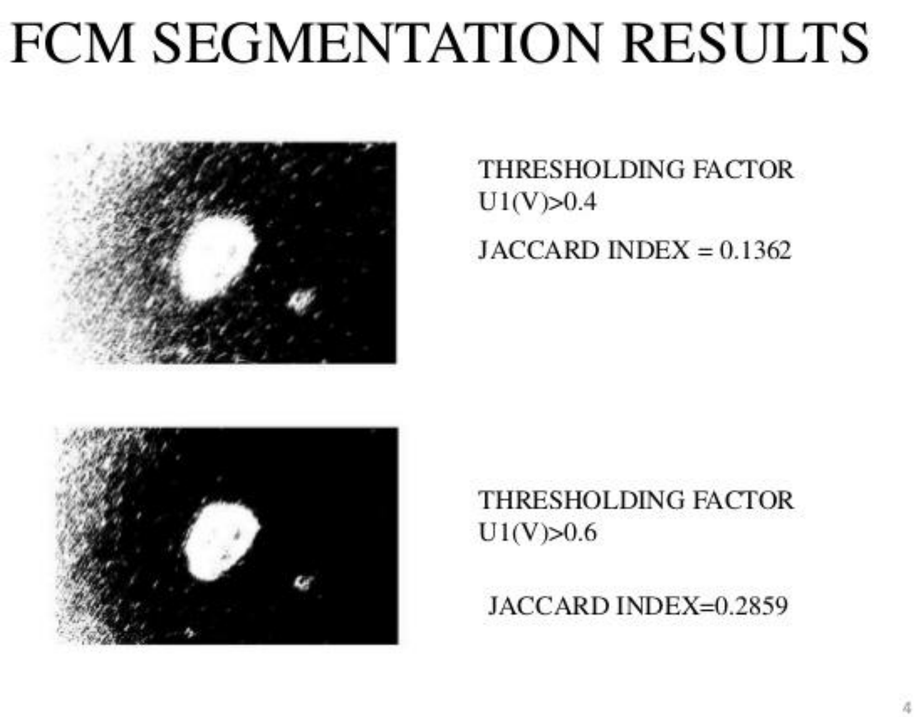


Figure 3.7 The FCM algorithm

The FCM algorithm generates Cluster by grouping pixels based on their color and proximity in the image plane. The original SLIC algorithm proposed by Achanta et al. [7,8] begins by considering for all operations a five-dimensional space, [labxy], where [lab] is the pixel color vector in the CIELAB color space while [xy] is the pixel position; also in the original algorithm the only required parameter is the number of Cluster (k) of similar size that we want to obtain. For this particular case, for convenience, we have started with the implementation of the SLIC algorithm carried out by Andrea Vedaldi [11] in which the transformation of the sRGB color space to CIELAB is not done and two important parameters are introduced, which are: the size nominal of each region (Cluster) that we define as regionTamanio; and the spatial regularization factor defined as factorReg; the image is then initially divided into a grid, with regular segments of size M * N.

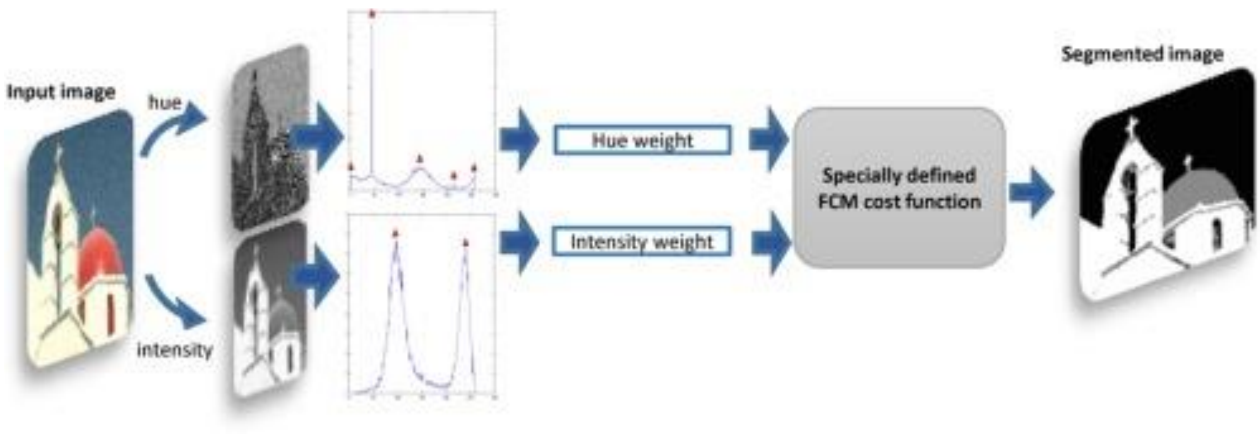


Figure 3.8 FCM minimization

The FCM algorithm was proposed by Dunn and modified by Bezdek. [9] The basic principle of FCM is the minimization of the following objective function: denote an image (or dataset) of N elements (pixels) that will be split into c cluster s. denote each center of the clusters. is the matrix of centers of cluster s. is the membership matrix (c * N) where -th pixel in the i-th cluster. One of the characteristics of the Cluster segmentation algorithms is that their adherence to the edges is limited to those with a high color gradient; In our particular case the SLIC algorithm, since it is based on hard partitions, will ignore all the fuzzy edges producing Cluster in which two or more different color shades appear. For this reason, once the Cluster have been obtained and all their elements (pixels) have been labeled, a statistical analysis of the variation of intensities within each cluster is carried out in order to identify which ones require deep segmentation to obtain greater adherence to the edges. To avoid errors caused by noise and other factors, this analysis is done through the luminosity histogram of each cluster.

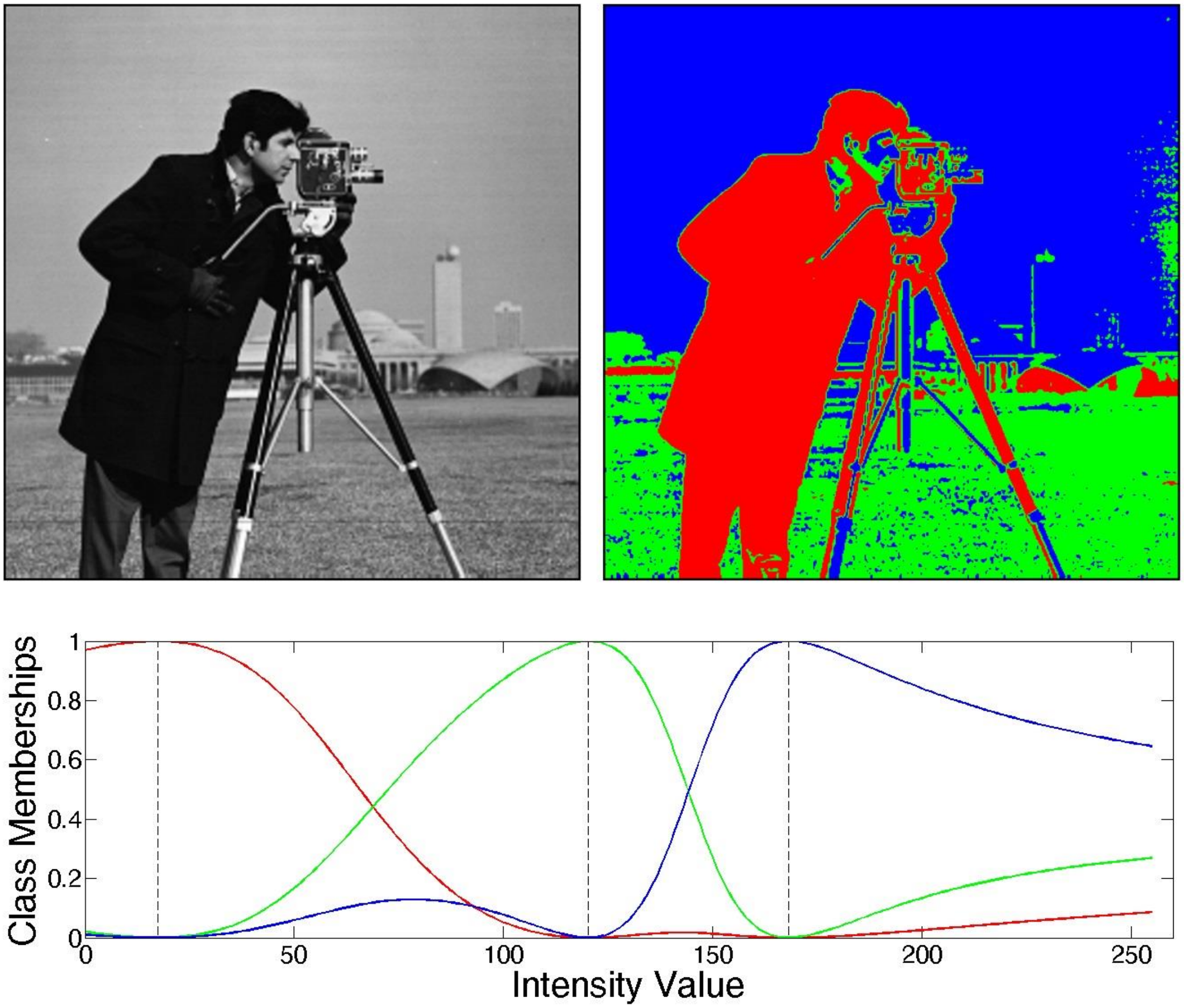


Figure 3.9 FCM Clusters

Clusters that visually display a variety of colors produce a luminosity histogram with a wide range that reflects high internal variation (Fig. 3.10; B, C, D). The clusters that present greater omogeneity, that is, represent a fragment of color intensity with low variation, produce histograms of reduced range The histogram range is easily calculated from the intensities present in the cluster shows the statistical analysis with the dispersion measures of the selected clusters based on the color of their member pixels; it is observed that the size of each Cluster (number of pixels, Npx) does not vary significantly; also, the smaller the Range the values of mean () median () and mode () are closer, indicating greater homogeneity; and finally the coefficient of variation (CV) indicates more clearly that the greater the presence of colors in the larger cluster is the CV. Special case are the clusters marked with labels D and E: their CV is almost identical, however cluster D, although in a very small fragment, it is visually heterogeneous; It is also clear that the range of cluster D is notably greater than that of cluster E. Considering these premises, the range of the histogram is considered as a measure of variation and is used as a parameter to determine to what degree deep segmentation will be performed.

3.4 PARTICLE SWARM OPTIMIZATION

Bio-inspired multi-target optimization algorithms have been shown to be a good tool for solving problems with different target functions [1]. The most representative algorithms are those that rely on evolution and also on swarms of particles. Specifically, algorithms based on the concept of evolution achieve a good approach to the Pareto front, however they require many generations.

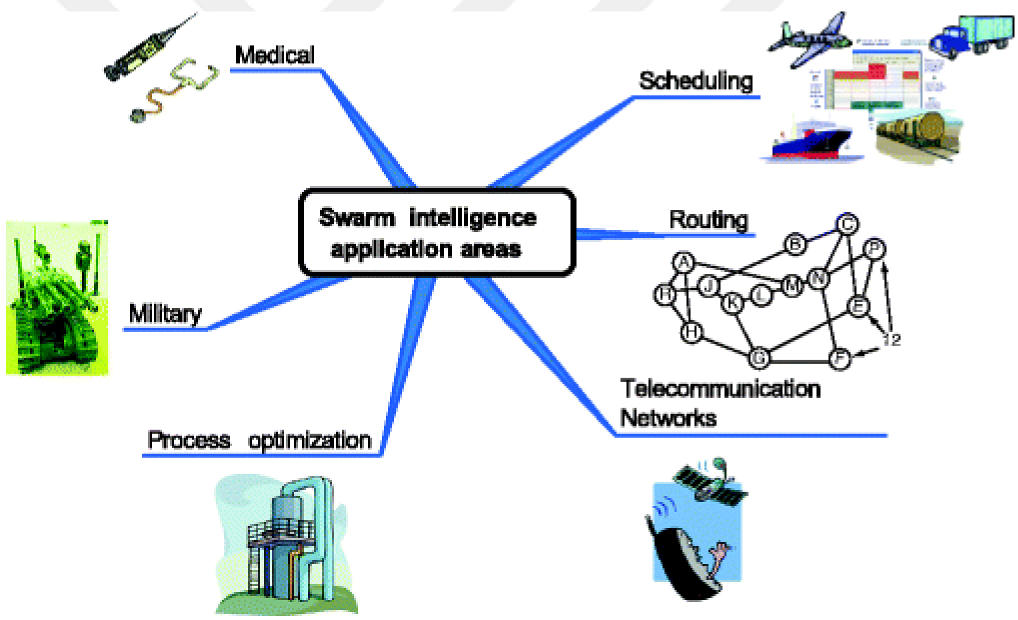


Figure 3.10 Applications of swarm algorithms

On the other hand, the algorithms that are based on swarms of particles present a high rate of convergence, however their main disadvantage is to achieve an adequate management of diversity [1], [2]. In the a priori method, the different objectives are combined in a cost function (scalar). In the progressive method, decision making and optimization are carried out at the same time. Partial selection (preference) information is provided as optimization is performed, providing a set of solutions to consider. For its part, the a posteriori method presents a set of optimal Pareto candidate solutions and chooses from that set [3]. The review is carried out observing the different approaches oriented to solve the multi-objective optimization problem and the algorithms developed. The review on Genetic Algorithm (GA) genetic algorithms is carried out in order to contextualize the different multi-objective evolutionary approaches, in addition to the development carried out on optimization algorithms based on particle swarms, Particle Swarm Optimization (PSO) applied to problems. multi-objective. Most of the algorithms considered use the concept of the Pareto front which consists of a set of solutions with the best compromises between the objectives to be optimized. With this approach, an external population is maintained for each generation, storing all the non-dominant solutions obtained to date. Each external population is mixed with the current population where its performance is assigned considering the number of solutions that a solution dominates. To ensure diversity between non-dominant solutions, the deterministic grouping technique is used [6], [18]. In this algorithm a father is generated by a mutation of a son. If the son dominates the father, the son is accepted as the next father and the iteration continues. If the father dominates the son, the son (offspring) is discarded and the new mutated solution is generated. If the offspring and the father are not dominant to each other, a comparison set of previously used individuals is employed. To keep the diverse population around the Pareto front, it is considered an archive of non-dominant solutions. A new offspring is compared to a file to verify if they dominate any member of the file, if so then the full offspring is accepted as a new parent. Key solutions are removed from the archive. In the case of the NSGA II algorithm, first the number of solutions that a given solution dominates as well as the set formed by the dominated solutions is established. With the previous information, the first front formed by the non-dominant solutions is obtained. From the current front there is a reduction in the dominated solutions. Non-dominant solutions that are permanent after this reduction are organized in a separate list. These processes continue to use the new front identified as the current front. A new population is formed considering the individual fronts from fronts to exceeding the population size. The solutions of the last allowed fronts are classified according to the ratio of the quantity of the population. NSGA II uses a parameter (called crowd distance) to estimate the density of each individual. The multitude distance of a solution is the average of the length of the side of the cube that encloses the point without including any other point in the population. The solutions of the last accepted front are classified according to the distance from the crowd. In this algorithm each solution can be encoded in more than one different alphabet [12], [13]. Additionally, the representation of a particular solution is not fixed, that is, the representation is adaptable and can be changed during the search process as an effect of the mutation operator.

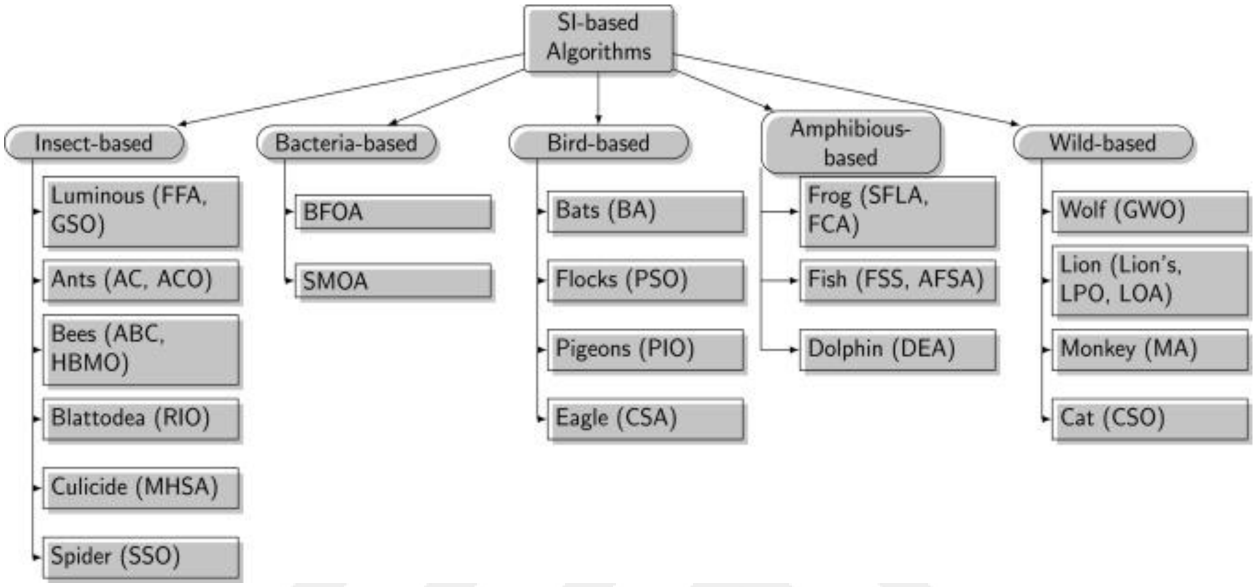


Figure 3.11 Types of Swarm intelligence SI algorithm

Each solution is represented by an integer corresponding to the number of elements in the alphabet and a string of alphabet symbols. If you have two symbols, it corresponds to the standard binary coding. The alphabet that each individual is encoded in may change during the search process. Each individual is selected for the mutation, which is the operator of unique variation. Children and parents are compared. If the sons dominate the father then the entire offspring of the new population and the father are removed. If the parents dominate the offspring after several mutations then another alphabet is chosen. A single population of individuals is used in this algorithm.

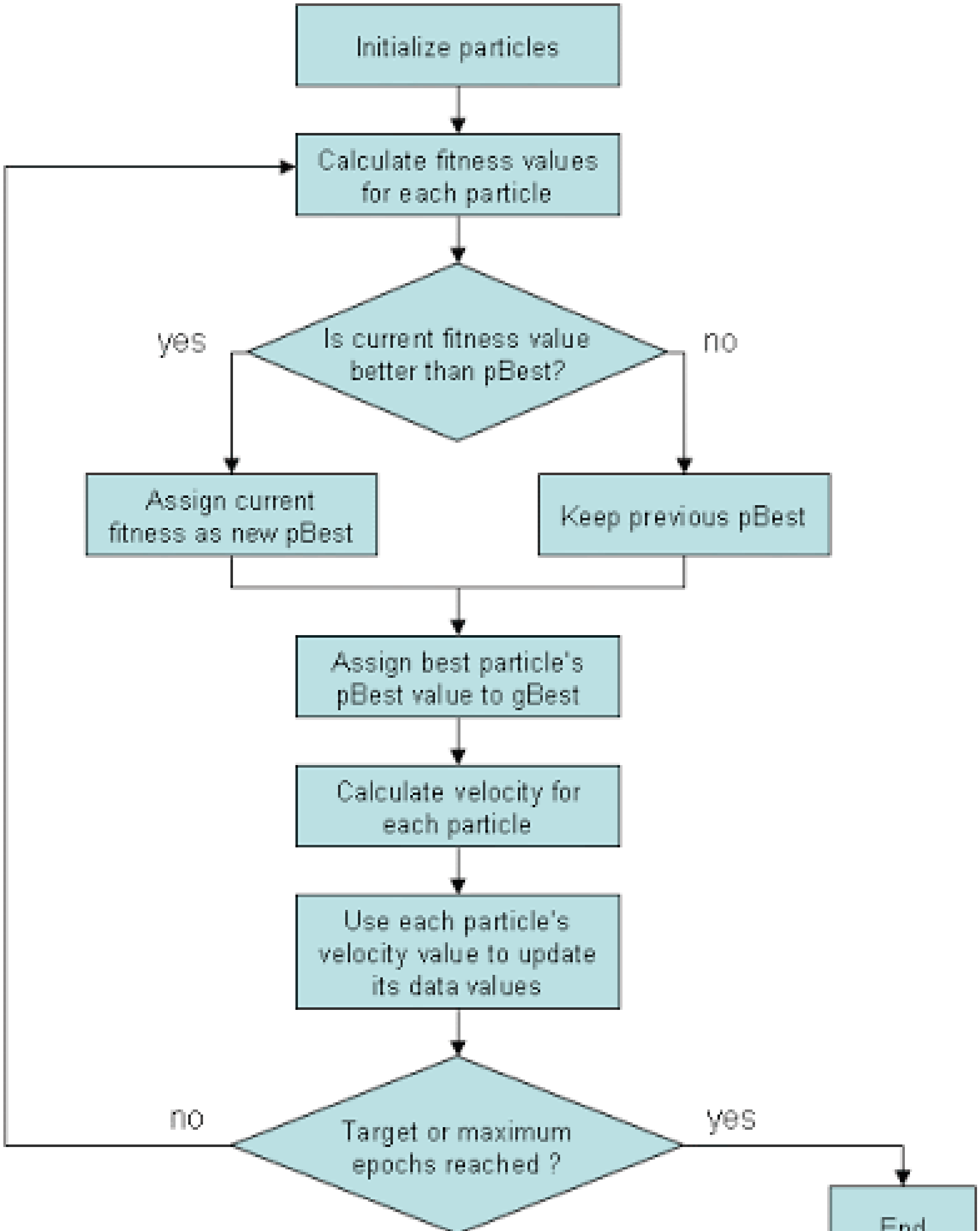


Figure 3.12 Diagram of PSO workflow

The description of the behavior of many living beings is characterized by presenting coordinated cooperative movement, such as flocks of birds, schools of fish and even microorganisms. One of the behaviors of interest is the circular movement of particles around a point called a vortex (Ebeling, 2002), since this form of locomotion can be a good strategy for foraging and evading obstacles and predators (García, 2007). Due to the above characteristics, these behaviors can be used in the proposal of optimization algorithms.

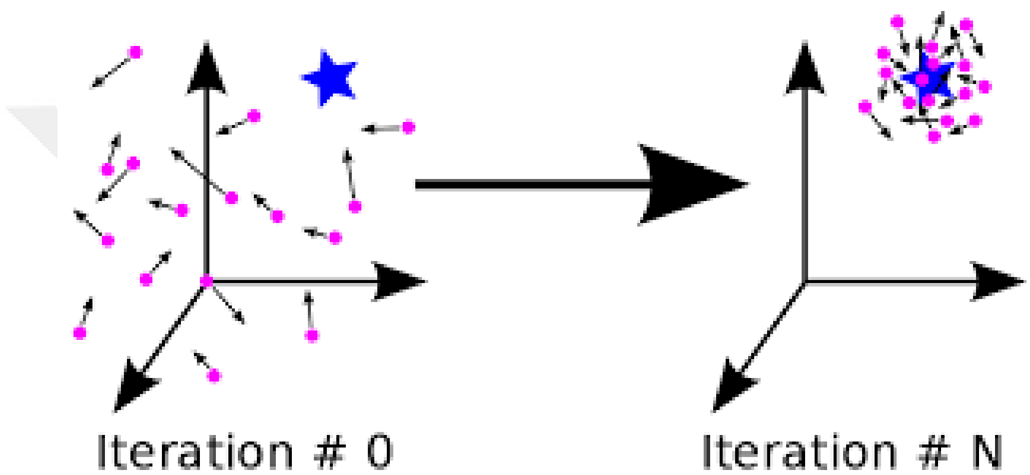


Figure 3.13 description of the behavior of PSO iteration

In nature you can see different behaviors which have been studied and represented analytically. In particular, congregations of individuals are an interesting topic because of the emerging behaviors that arise (Sumpter, 2006). Among the works that can be highlighted is the one presented by Vicsek (1995), where a basic model is developed to represent a swarm of individuals. This last author carries out an extension of his work (Vicsek, 2008), where he describes some representative patterns of swarms. On the other hand, Sumpter (2006) reviews collective behavior for swarming, observing self-regulating properties and principles of collective behavior such as: integrity, variability, positive feedback, negative feedback, response thresholds, direction, inhibition, redundancy, timing and selfishness. On the different approaches considered for swarm models, Couzin (2005) analyzes the effect that the leadership of an individual has, Bajec (2009) considers the different forms of organization that birds present and Zhang (2008) observes the effect that incorporating prediction mechanisms in a swarm model. Particle models with vorticity behavior Vorticity is a behavior that is pre-coupling that exists between inertial forces and viscous forces (

Reynolds number to is performed using the Navier Stokes equations, which are often difficult to solve analytically in general cases Çengel teriza [14] by the movement of form rotational particle around a point, which is behavior occurs in swarms individuals such as fish, birds and bacteria, among others. There are two models that are the most used to represent this behavior, one consists of the self-propelled particle model (Levine, 2000; D'Orsogna, 2006) and the other corresponds to that of Brownian active particle (Ebeling, 2006). The latter, unlike the former, considers a stochastic component. These models usually use pot Morse features to represent the interaction between individuals; However, in Erdmann's work (2005) a model can be observed that uses a parabolic potential Of the different bioinspired optimization algorithms, according to Bratton (2007) the particle swarm optimization (PSO) algorithms of particle swarms are a good alternative; However, they tend to present early convergence in local minima (Evers, 2009; Schutte, 2002), and additionally, they are susceptible to a poor selection of their parameters, as shown by Hvass (2010) and Van den Bergh (2001). Due to the above, local minimums have been developed and evaded, having a better exploration of the solution space.

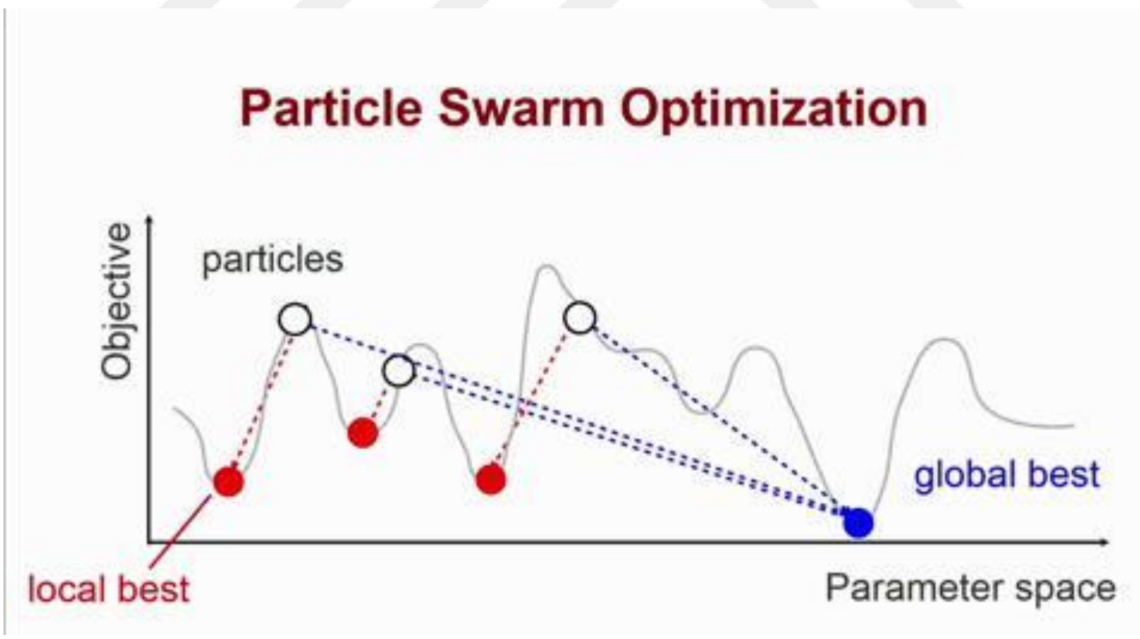


Figure 3.14 Parameter space of PSO

A general strategy for escaping local minima is to carry out a dispersion (explosion) process, after converging to a local minimum. An example of this concept can be seen in Mesa's (2010) work with the Supernova optimization algorithm, in Krishnanand's (2009) work for the Glowworm optimization algorithm and in Passino's (2005) work with the optimization based on foraging of bacteria.

In particular, for the PSO algorithm a modulated inertia primes as proposed by Feng (2007) and Yin (2009). This approach seeks to control the exploration of the algorithm on the search space. The aforementioned authors state that a large inertia factor accelerates the convergence while a small value improves the functional of the PSO algorithm consists of restarting it when it is considered that there is a stagnation of it (García, 1997).

In addition to the modulated inertia approach, Hendtlass (2005) proposes a method called Waves of Swarm Particles WoSP, which seeks to boost the swarm so that it can escape from a local minimum and thus continue the process. of exploration. On the other hand, Parsopoulos (2004) proposes to use repulsion techniques for each local minimum found, thus hoping to avoid previously found solutions. Likewise, Liang (2006) presents a variant of the PSO algorithm called integral learning, which consists of using the historical information of the particles to update their speed. This approach seeks to preserve the diversity of the swarm by avoiding premature convergence.

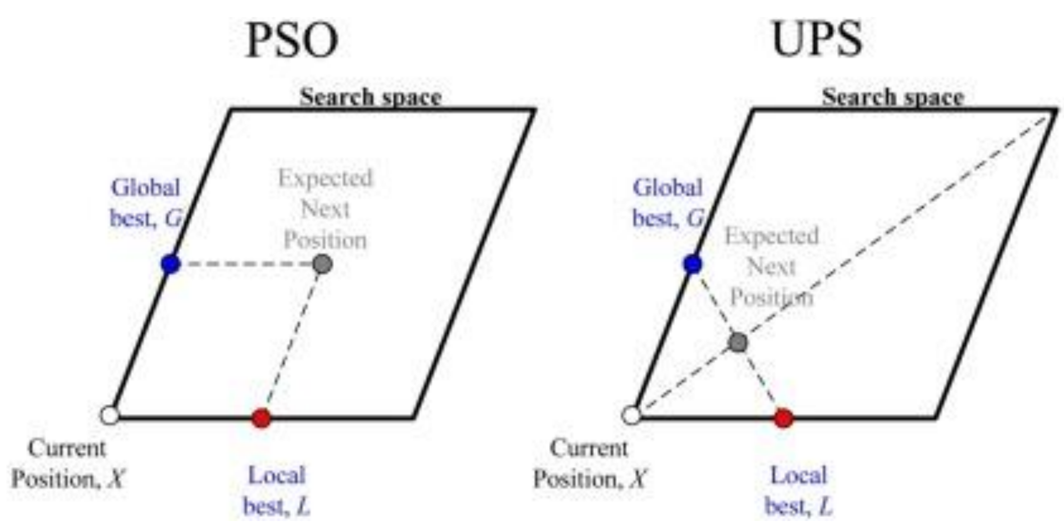


Figure 3.15 PSO vs UPS

Finally, among the optimization algorithms that use the concept of vorticity is the one presented by Menser (2006), where an algorithm based on behavior is developed # it is called Particle Swirl Algorithm (PSA). re of the proposal made in this document, which seeks to use the concept of vorticity to ensure that the swarm of particles escapes from a local minimum and thus can continue the search process. A first work where an optimization algorithm based on swarm behavior with ca who does not consider the individual search to improve the scanning characteristics of the swarm is proposed. Previously, this same concept was used for planning mobile robot trajectories (Espitia, 2011a, 2011b).

3.5 KNN CLASSIFICATION

The k-NN algorithm is one of the best-known classification algorithms and an example of supervised learning. In this algorithm we will use a set of already classified examples that we will call a training set or train to classify the new examples. A new model is not created, but the model is the train set itself.

The algorithm is called k-NN (k-Nearest Neighbors) because it classifies each new example by calculating the distance of that example with all the ones in the train set. The predicted class for this new example will be given by the class to which the closest examples of the train set belong, the value of k is the one that determines how many neighbors we must look to predict the class. Thus, with a value of k = 1, the predicted class for each new example will be the class to which the closest example of the train set belongs.

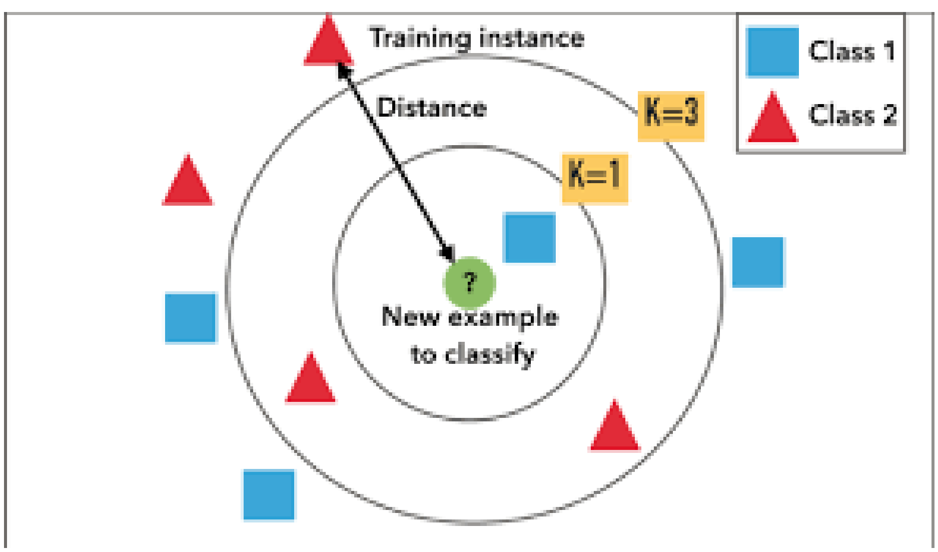


Figure 3.16 KNN classification

The main problem of the k-NN algorithm is finding the value of k with which we obtain a greater performance when classifying, generally a technique known as cross validation is used. This technique consists of dividing the train set into different parts, for example into 5 and classifying each of those parts using k-NN with the desired k values, using the other 4 parts as the train set. In this way we will have the performance of k-NN with all the values of k for each of the 5 partitions that we have made of the original train set. By averaging these returns, we would obtain the value of k with which the best return is obtained and we would use it to classify any new example.

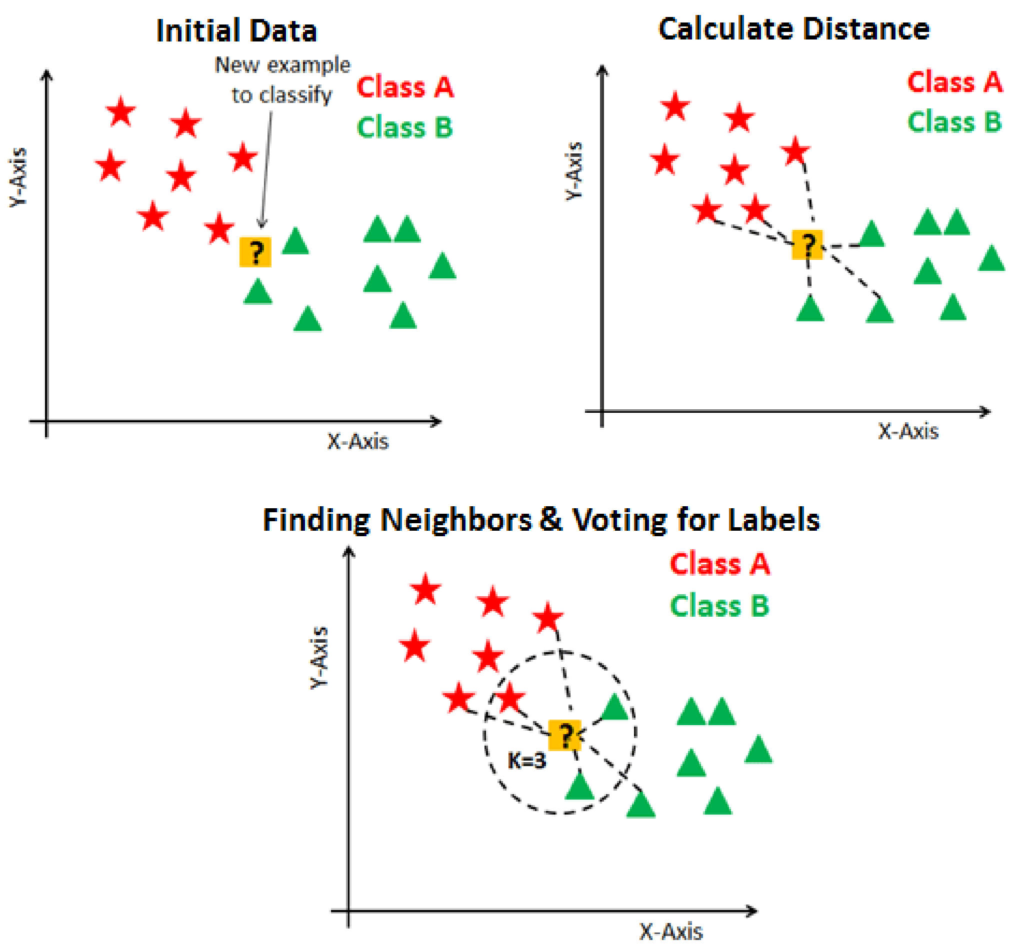


Figure 3.17 KNN voting

However, there is another technique that does not classify each new example with the same value of k, but the k with which we will classify a new example will be determined based on different parameters such as which are its closest neighbors. The project that we are going to carry out uses this other technique.

This method that we are going to implement consists of obtaining a local k value associated with each example of the train set (from now on we will call each example of the train set prototype). The method learns this value from the train set itself. For each prototype, we evaluate the performance of k-NN with all the values of k in an interval [kmin, kmax] and choose the one with the best value.

The approach of the method is the following, we will assign to each prototype a value of k, whose ideal value will be the optimal number of neighbors to take into account to classify an example that is in its neighborhood (that an example is in the neighborhood of a prototype means that prototype is your closest neighbor.) In the standard k-NN method the prototypes are of the form:

(x1, ..., xn, y)

Where x1, ..., xn are the attributes of the prototype and y is the class to which it belongs. However, with this method we will modify the prototypes so that they are of the form: (x1,…, xn, y, k)

Where k will be the local value of k associated with the prototype that we will use to classify the examples from its neighborhood.

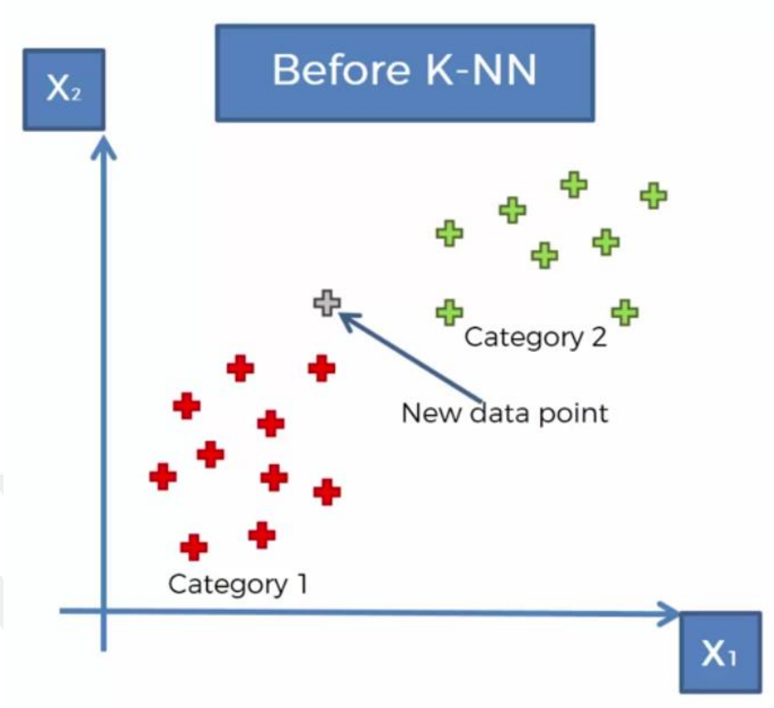


Figure 3.18 KNN categories

To obtain the optimal local value of k for each prototype we will need 2 different measurements:

• Global Accuracy: this measurement will be calculated by running the standard k-NN algorithm on the entire train set, with all values of k from 1 to kmax. To do this, we will cross-validate 10 partitions on the train set and run the standard k-NN algorithm to classify each of those 10 partitions using the other 9 as a classifier. With this, we will obtain the precision of the k-NN algorithm in each of the 10 partitions, making the average we will obtain a vector that we will call Accglobal that will be an estimate of the precision of the standard k-NN algorithm with all the values of k for the train set.

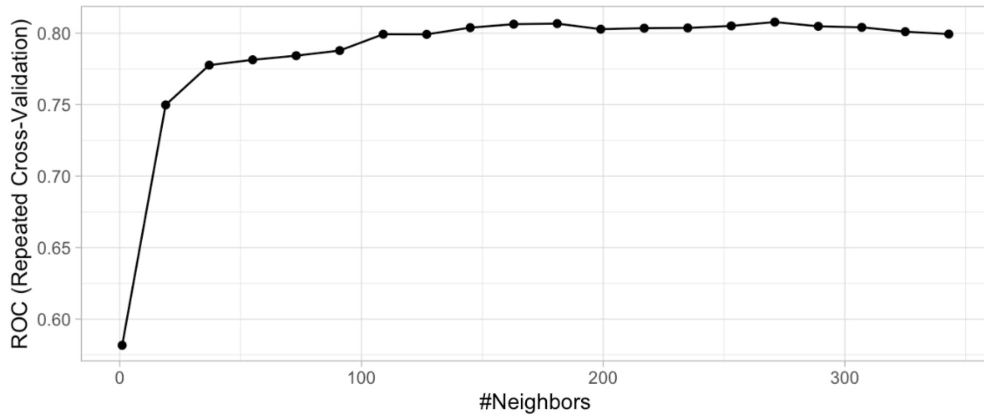


Figure 3.19 KNN neighbors

• Local Accuracy: to calculate this measurement we will also run the standard k-NN algorithm on the entire train set, but this time we will look at the effect of each value of k from 1 to kmax in the neighborhood of each prototype. In principle, the neighborhood of each prototype are only those prototypes for which this is the closest neighbor. However, some prototypes are not the closest neighbor

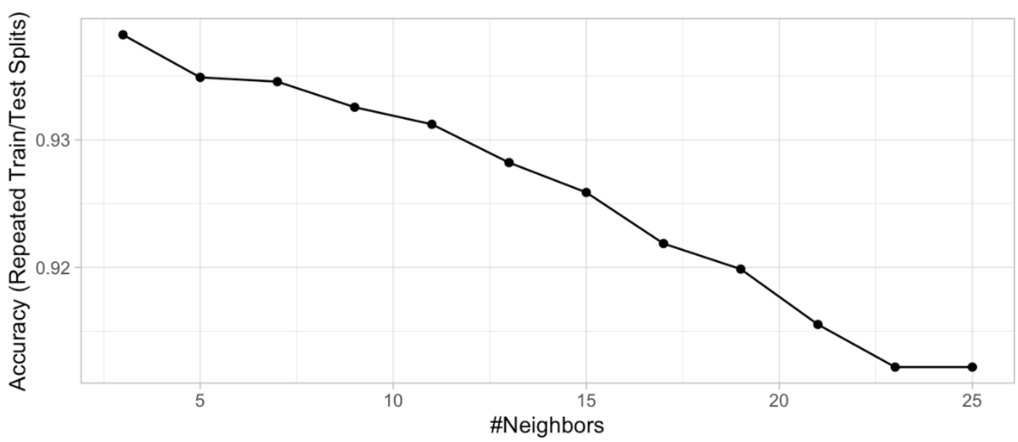


Figure 3.20 KNN Accuracy

from no other or very few. To avoid the negative effect that there are very few prototypes in a neighborhood, the neighborhood of each prototype will be formed by those prototypes for which this is one of the 3 closest neighbors. This is how we ensure larger neighborhoods.

The result will be an n-prototype x kmax matrix that we will call accLocal, where we will have the precision of the standard k-NN over the neighborhood of each prototype with all the values of k.

Now we have a measure that contains the precision of the k-NN algorithm over the entire train set and another that contains the precision of the k-NN algorithm over the neighborhood of each prototype. We need to take both measures into account to obtain the optimal value of k that will be associated with each prototype, so we will calculate a new measure called eval where we will have both measures. For this, to the precision that we had of the k-NN on each prototype we will add the precision of the k-NN over the entire training set, that is, to each row of the acclocal matrix we will add the accglobal vector:

eval = acclocal + accglobal

This new eval measure will be an n-prototype x kmax matrix, where we will have the total precision of the k-NN algorithm over the entire train set. Each row represents a prototype, therefore, the position with the highest precision value of each row will be the local value of k associated with that prototype.

With the local value of k already calculated and added to each prototype, it would only remain to classify new examples. We calculate for each new example the distance to all the prototypes and classify the example taking into account as many neighbors as indicated by the local k of its closest prototype.

4. PROPOSED METHOD

We use the 3D environment of the Autodesk Maya software to set up the 3D environment and import the models into MATLAB to simulate the robot's movement, the robot will move and take an image every 5 units of distance (in our case centimeters) depending on the global scale of the environment, the FCM algorithm will segment that image and the KNN will classify these segments into Safe and unsafe regions, the figure below shows the environment set up for the 3D simulator:

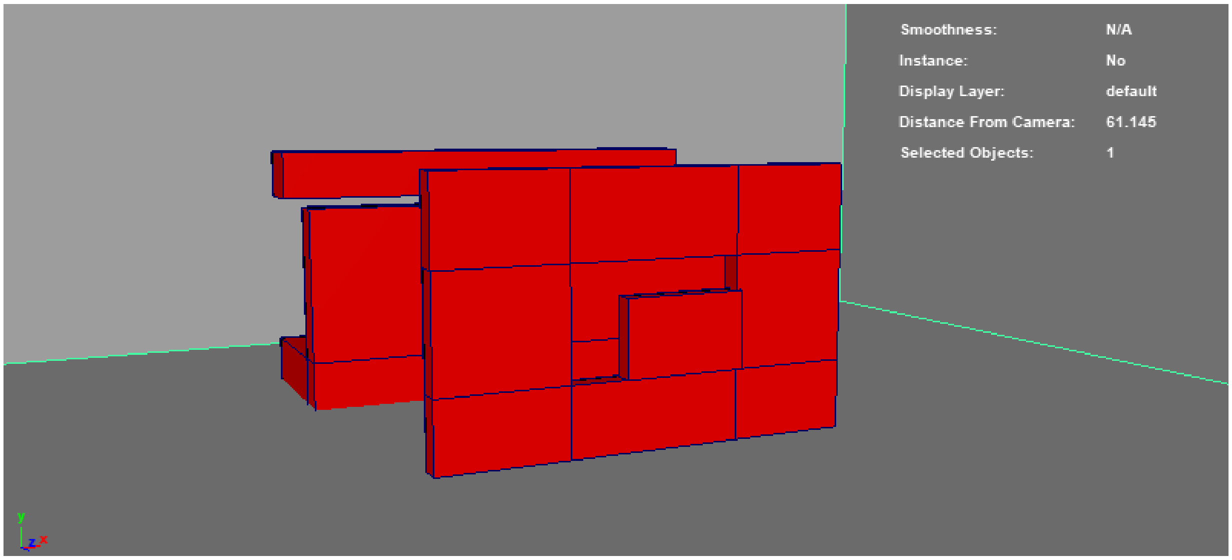


Fig.4.1 Environment set up in Autodesk Maya

While the figure below figure 5 shows the image taken from the front view of the camera, and the figure 6 shows the segmentation results of the image:

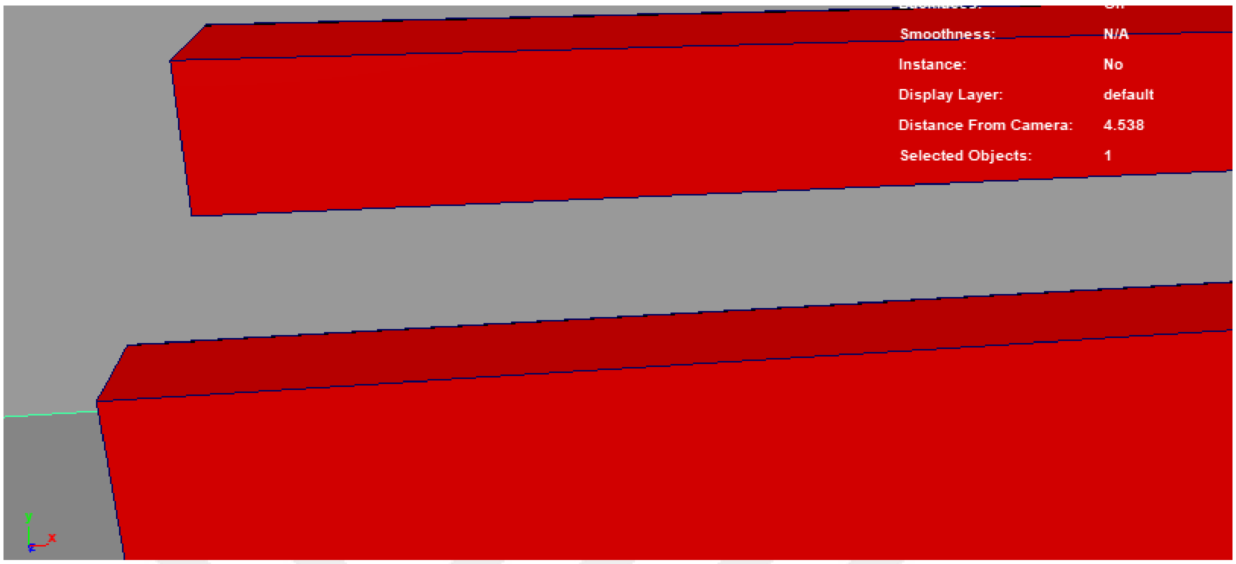


Fig.4.2 Front view camera image

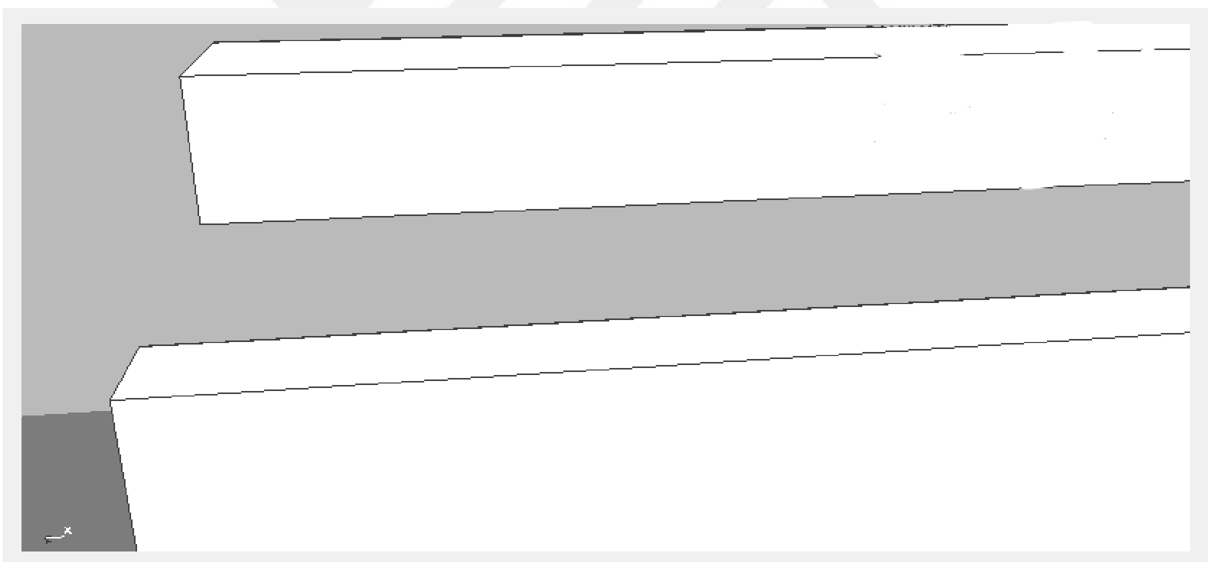


Fig.4.3 FCM segmentation of the image

The simulation result show maximum accuracy in path planning and avoiding the obstacles in the way of the robot:

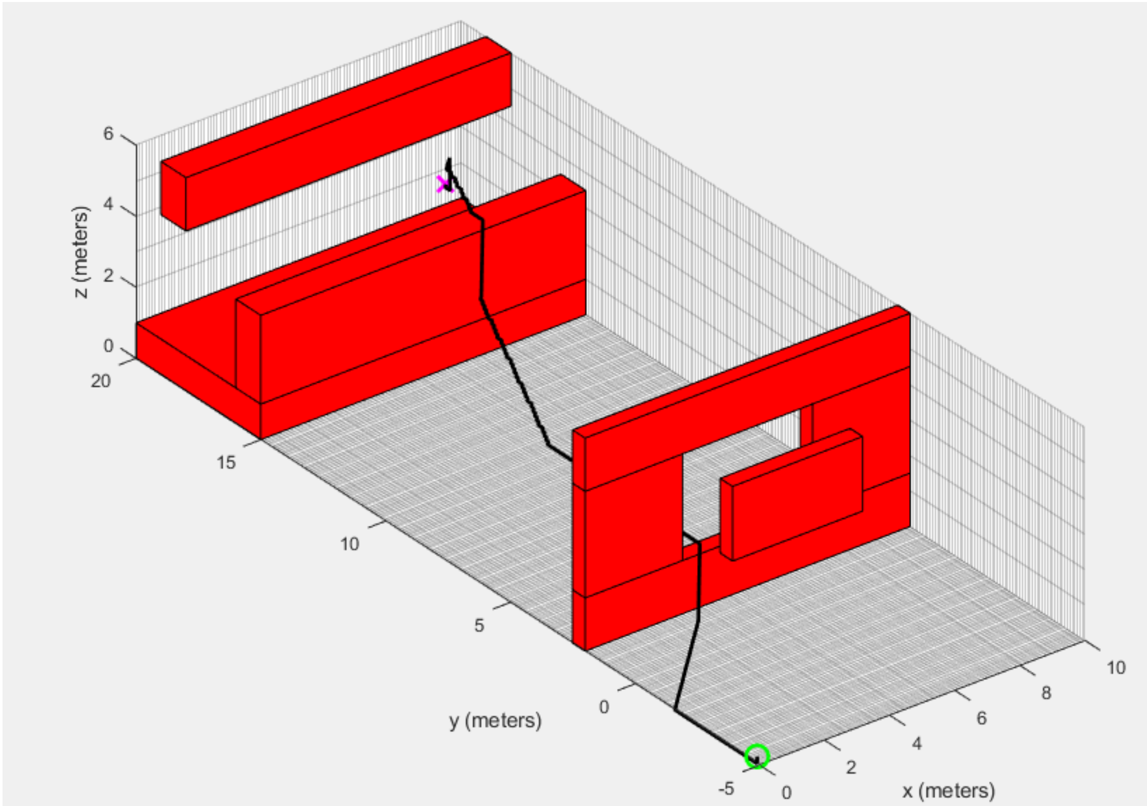


Fig.4.4 Proposed robot path planning simulation

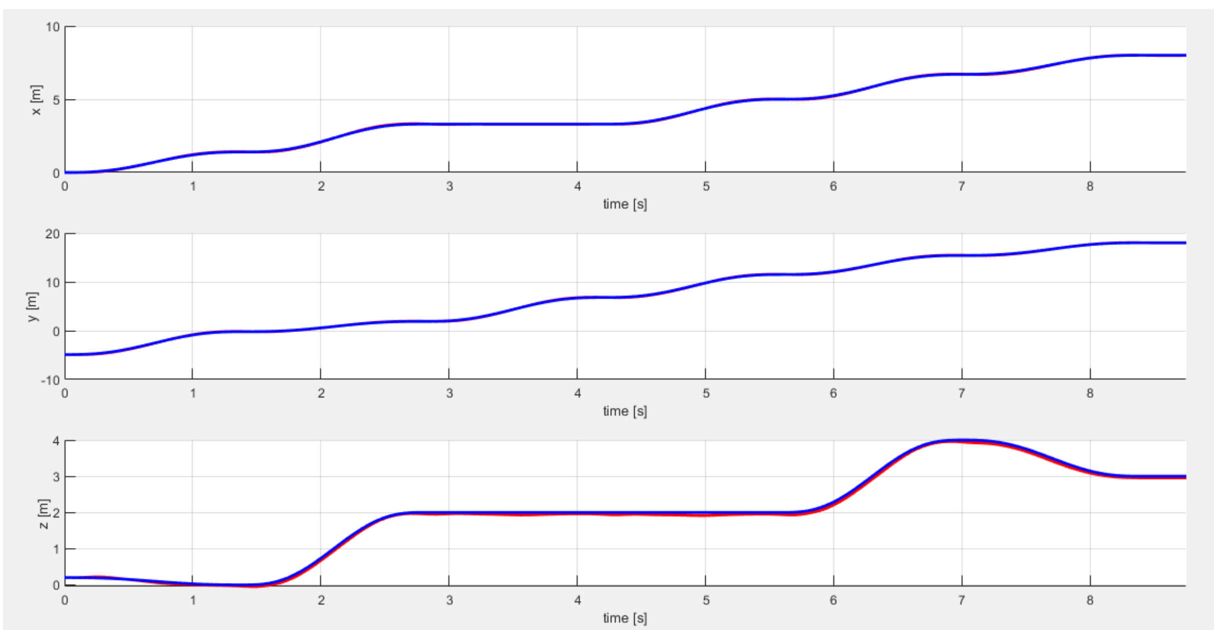


Fig.4.5 The change in X, Y, Z position of the mobile robot

Table 1 shows the MSE and PSNR of the fuzzy c-means algorithm compared to other methods of segmentation like K-means, Region growing (RG), Hierarchical K-Means (HKM) and Multiway segmentation (MWS) :

Table-1 MSE and PSNR of the FCM algorithm compared to other common methods of segmentation

Metric	K-means	FCM	RG	HKM	MWS
PSNR	66.73	67.99	52.53	54.91	54.99
MSE	0.33	0.19	0.25	0.39	0.22

While the figure below figure 8 shows that the KNN although it might not give the best accuracy compared to other classification method but it definitely produces the best time in training and decision making

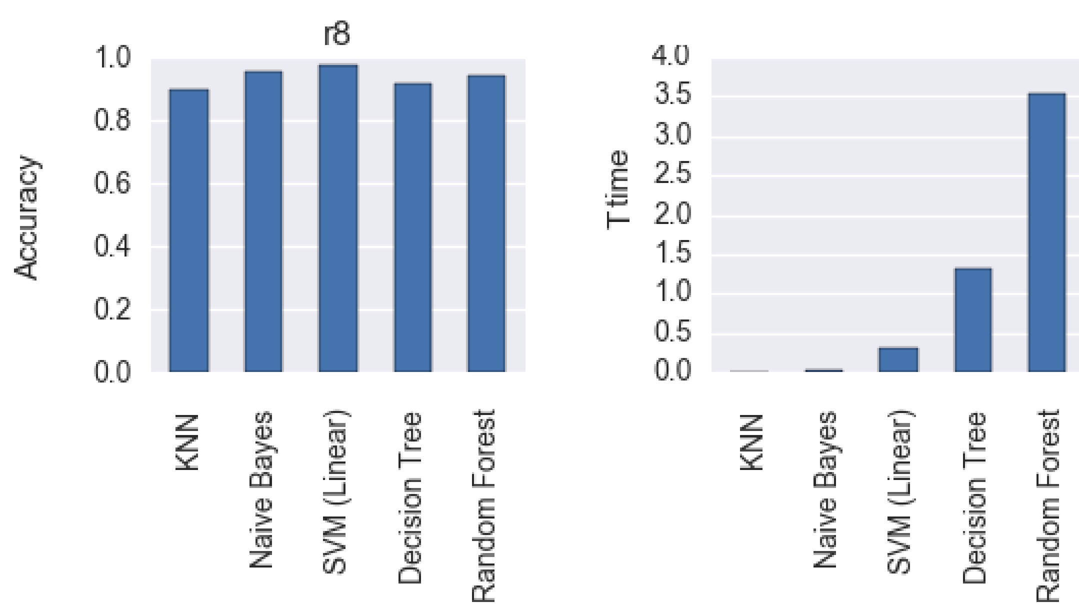
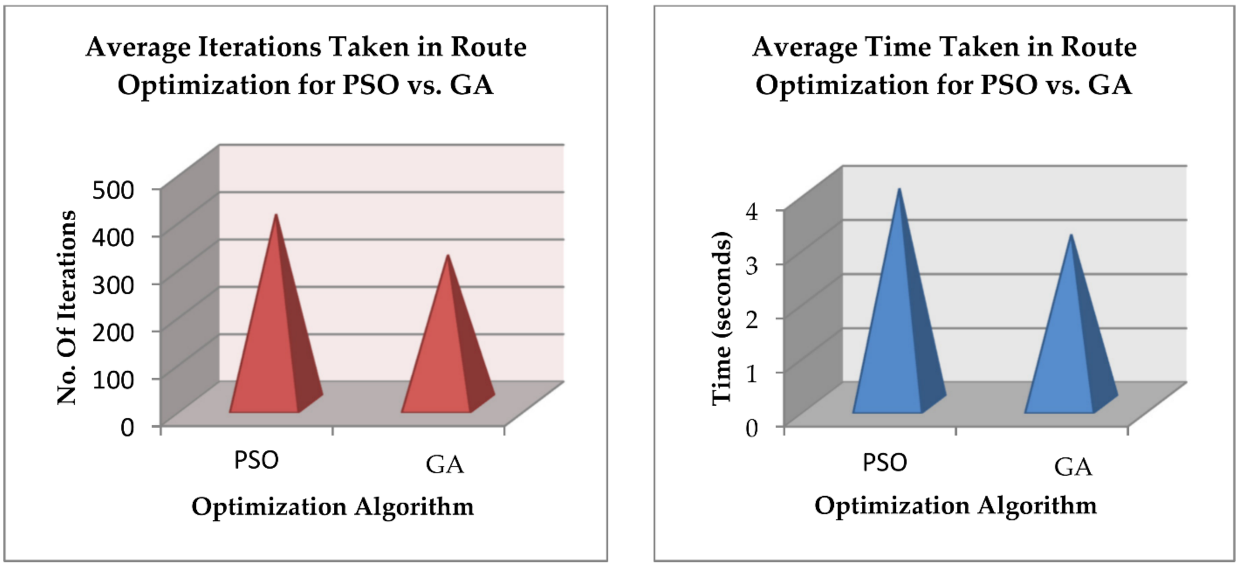


Fig.4.6 KNN accuracy and time compared to other methods implemented in the same situation

As for the decision making in PSO algorithm we compare the Accuracy and time to the genetic algorithm or GA as shown in the figure below



(a) (b)

Figure 4.7 PSO accuracy and time compared to the Genetic algorithm

5. CONCLUSION

In this work we presented a scheme for segmenting the robot rea using FCM algorithm and a PSO algorithm for planning the path between these areas, these algorithms are a simple and efficient method for planning paths. In the case of variable potential fields, it is necessary to follow the cells with smaller numerical values to find the arrival point, while in SI algorithms the path depends on the function evaluation. The coding used for the PSO algorithm only allows working in a square work area, in the case of potential fields it is indifferent to the shape of the terrain, which makes it more robust. It is possible to execute the two strategies for static environments or for dynamic environments, the only thing that changes are the update of the data that can be obtained after obtaining the data with a camera. The number of obstacles depends on the size of the track, in addition to the risk areas they can be increased or decreased depending on the shape of the obstacle. The implemented vision algorithm allows the detection of movement of up to 6 mobile robots on the scene in real time for the application. Because the response time of the vision algorithm was sometimes less than that of the implemented control algorithms, it was necessary to update the data at the speed given by the strategies, which meant that the speed of the mobile robot was moderate.

REFERENCES

[1] C. Coello, D. Van Veldhuizen, G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, Second Edition, 2007.

[2] C. Coello., M. Salazar, *MOPSO: A proposal for multiple objective particle swarm optimization*, In Proceedings of the IEEE Congress of Evolutionary Compututation, 2002.

[3] D. Van Veldhuizen, G. Lamont, "Multiobjective Evolutionary Algorithms: Analyzing the Stateof-the-Art", *Evolutionary Computation*, vol. 8, no. 2, pp. 125-147, 2000.

[4] J. Schaffer, *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 93-100, 1985.

[5] D. Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Ph.D. thesis, Air Force Institute of Technology, Wright - Patterson AFB, Ohio, 1999.

[6] E. Zitzler, L. Thiele, *An evolutionary algorithm for multiobjective optimization: The strength Pareto approach*, Technical report 43, Computer engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 1999.

[7] E. Zitzler, L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach", *IEEE Transaction on Evolutionary Computation*, vol 3, no. 4, pp. 257-271, 1999.

[8] J. Knowles, D. Corne, *The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization*, IEEE Congress on Evolutionary Computation (CEC), 1999.

[9] D. Corne, J. Knowles, M. Oates, "The Pareto envelope - based selection algorithm for multiobjective optimization", *Parallel Problem Solving from Nature - PPSN VI*, pp. 839-848, 2000.

[10] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, "A fast elitist non - dominated sorting genetic algorithm for multi-objective optimization: NSGA II", *Parallel Problem Solving From Nature - PPSN VI*, pp. 849-858, 2000.

[11] E. Zitzler, M. Laumanns, L. Thiele, *SPEA 2: Improving the Strength Pareto Evolutionary algorithm*, Technical report 103, Computer engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2001.

[12] D. Dumitrescu, C. Grosan, M. Oltean, *Simple Multiobjective, Evolutionary Algorithm, Seminars on Computer Science,* Faculty of Mathematics and Computer Science, Babe-Bolyai University of Cluj-Napoca, pp. 3-12, 2001.

[13] D. Dumitrescu, C. Grosan, M. Oltean, "A new evolutionary adaptive representation paradigm", *Studia Universitas Babes-Bolyai, Seria Informatica*, vol. XLVI, no. 1, pp. 15-30, 2001.

[14] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[15] N. Srinivas, K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms", *Journal of Evolutionary Computation,* vol. 2, no. 3, pp. 221-248, 1994.

[16] J. Horn, N. Nafpliotis, D. Goldberg, *A niched Pareto genetic algorithm for multiobjective optimization*, IEEE World Congress on Computational Intelligence, IEEE Conference on Evolutionary Computation, 1994.

[17] M. Erickson, A. Mayer, J. Horn, "The niched Pareto genetic algorithm 2 applied to the design of groundwater remediation systems", *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, vol. 1993, pp.
681-695, 2001.

[18] E. Zitzler, M. Laumanns, L. Thiele, "SPEA 2: Improving the Strength Pareto Evolutionary Algorithm", *CIMNE, Evolutionary Methods for Design, Optimisation, and Control*, pp. 95-100, 2002.

[19] D. Cvetkovic, I. Parmee, "Preferences and their application in evolutionary multiobjective optimization", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 42-57, 2002.

[20] A. Lara, G. Sanchez, C. Coello, O. Schutze, "HCS: A New Local Search Strategy for Memetic Multiobjective Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, 2010.

[21] Y. Junchi, L. Guoqiang, *Double space based multiobjective evolutionary algorithm*, International Conference on Machine Learning and Cybernetics (ICMLC), 2012.

[22] P. Bosman, “On Gradients and Hybrid Evolutionary Algorithms for Real-Valued Multiobjective Optimization”, *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, 2012.

[23] D. Brockhoff, E. Zitzler, *Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods*, IEEE Congress on Evolutionary Computation (CEC), 2007.

[24] I. Karahan, M. Köksalan, “A Territory Defining Multiobjective Evolutionary Algorithms and Preference Incorporation”, *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, 2010.

[25] Z. Qingfu, L. Hui, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, 2007.

[26] L. Chi-Ho, K. Ye-Hoon, K. Jong-Hwan, *Multiobjective evolutionary algorithm reinforcing specific objective*, IEEE Congress on Evolutionary Computation (CEC), 2008.

[27] L. Hai-lin, L. Xueqiang, *The multiobjective evolutionary algorithm based on determined weight and sub-regional search*, IEEE Congress on Evolutionary Computation (CEC), 2009.

[28] G. Montemayor-Garcia, G. Toscano-Pulido, *A study of surrogate models for their use in multiobjective evolutionary algorithms*, 8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE), 2011.

[29] Z. Aimin, Z. Qingfu, Z. Guixu, *A multiobjective evolutionary algorithm based on decomposition and probability model*, IEEE Congress on Evolutionary Computation (CEC), 2012.

[30] L. Hai-Lin, W. Dan, *A constrained multiobjective evolutionary algorithm based decomposition and temporary register*, IEEE Congress on Evolutionary Computation (CEC), 2013.

[31] W. Chen, Y. Gong, Z. Zhan, J. Zhang, Y. Li, Y. Tan, “An Evolutionary Algorithm with Double-Level Archives for Multiobjective Optimization”, *IEEE Transactions on Cybernetics*, vol.

PP, no. 99, 2014.

[32] W. Rui, R. Purshouse, P. Fleming, “Preference-Inspired Coevolutionary Algorithms for Many-Objective Optimization”, *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474-494, 2013.

[33] Q. Rongbin, D. Wenli, W. Zhenlei, Q. Feng, *Multiobjective evolutionary algorithm based on the Pareto Archive and individual migration*, 7th World Congress on Intelligent Control and Automation (WCICA), 2008.

[34] J. Fieldsend, R. Everson, S. Singh, "Using unconstrained elite archives for multiobjective optimization", *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 305-323, 2003.

[35] H. Shinn-Ying, S. Li-Sun, C. Jian-Hung, "Intelligent evolutionary algorithms for large parameter optimization problems", *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 6, pp.
522-541, 2004.

[36] H. Sato, H. Aguirre, K. Tanaka, *Local dominance using polar coordinates to enhance multiobjective evolutionary algorithms*, Congress on Evolutionary Computation, 2004.

[37] H. Sato, H. Aguirre, K. Tanaka, *On the locality of dominance and recombination in multiobjective evolutionary algorithms*, IEEE Congress on Evolutionary Computation, 2005.

[38] W. Yong, C. Zixing, *A constrained optimization evolutionary algorithm based on multiobjective optimization techniques*, IEEE Congress on Evolutionary Computation, 2005.

[39] M. Sato, H. Aguirre, K. Tanaka, *Effects of d-Similar Elimination and Controlled Elitism in the NSGA-II Multiobjective Evolutionary Algorithm*, IEEE Congress on Evolutionary Computation (CEC), 2006.

[40] C. Zixing, W. Yong, "A Multiobjective Optimization-Based Evolutionary Algorithm for Constrained Optimization", *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, 2006.

[41] L. Rachmawati, D. Srinivasan, "Multiobjective Evolutionary Algorithm With Controllable Focus on the Knees of the Pareto Front", *IEEE Transactions on Evolutionary Computation*, vol.
13, no. 4, 2009.

[42] Y. Song, J. Ji, Y. Wang, C. Liu, *A New Evolutionary Algorithm for Solving Multiobjective Optimization*, Fifth International Conference on Natural Computation (ICNC), 2009.

[43] D. Phan, J. Suzuki, *Boosting Indicator-Based Selection Operators for Evolutionary Multiobjective Optimization Algorithms*, 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2011.

[44] C. Chi, Y. Shiu, "A Multiobjective Evolutionary Algorithm That Diversifies Population by Its Density", *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, 2012.

[45] L. Ke, A. Fialho, S. Kwong, Z. Qingfu, "Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition", *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, 2014.

[46] K. Parsopoulos, M. Vrahatis, *Multi-Objective Particles Swarm Optimization Approaches*, IGI Global, 2008.

[47] K. Parsopoulos, M. Vrahatis, *Particle swarm optimization method in multiobjective problems*, In Proceedings of the ACM Symposium on Applied Computing, 2002.

[48] U. Baumgartner, C. Magele, W. Renhart, "Pareto optimality and particle swarm optimization", *IEEE Transactions on Magnetics*, vol. 40, no. 2, pp. 1172-1175, 2004.

[49] M. Mahfouf, M. Chen, D. Linkens, "Adaptive weighted particle swarm optimisation for multi-objective optimal design of alloy steels", *Lecture notes in computer science*, vol. 3242, pp. 762-771, 2004.

[50] X. Li, "A non-dominated sorting particle swarm optimizer for multi-objective optimization", *Lecture notes in computer science*, vol. 2723, pp. 37-48 2003.

[51] X. Hu, R. Eberhart, *Multi-objective optimization using dynamic neighborhood particle swarm optimization*, IEEE Congress Evolutionary Compututation, 2002.

[52] X. Hu, R. Eberhart, Y. Shi, *Particle swarm with extended memory for multi-objective optimization*, IEEE Swarm Intelligence Symposium, 2003.

[53] K. Parsopoulos, M. Vrahatis, *On the computation of all global minimizers through particle swarm optimization*, IEEE Transactions on Evolutionary Computation, 2004.

[54] C. Chow, H. Tsui, *Autonomous agent response learning by a multi-species particle swarm optimization*, IEEE Congress on Evolutionary Computation, 2004.

[55] J. Fieldsend, S. Singh, *A multiobjective algorithm based upon particle swarm optimisation, An efficient data structure and turbulence*, In Proceedings of the UK Workshop on Computational Intelligence, 2002.

[56] T. Ray, K. Liew, "A swarm metaphor for multi-objective design optimization", *Engineering Optimization*, vol. 34. no. 2, pp. 141-153, 2002.

[57] T. Bartz-Beielstein, P. Limbourg, J. Mehnen, K. Schmitt, K. Parsopoulos, M. Vrahatis, *Particle swarm optimizers for Pareto optimization with enhanced archiving techniques*, IEEE Congress on Evolutionary Computation, 2003.

[58] D. Srinivasan, T. Seow, *Particle swarm inspired evolutionary algorithm (PSEA) for multi-objective optimization problem*, IEEE Congress on Evolutionary Computation, 2003.

[59] S. Mostaghim, J. Teich, *Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)*, IEEE Swarm Intelligence Symposium, 2003.

[60] X. Li, "Better spread and convergence: Particle swarm multi-objective optimization using the maximin fitness function", *Lecture notes in computer science*, vol. 3102, pp. 117-128, 2004.

[61] G. Toscano, C. Coello, "Using clustering techniques to improve the performance of a particle swarm optimizer", *Lecture notes in computer science*, vol. 3102, pp. 225-237, 2004.

[62] M. Reyes-Sierra, C. Coello, *Online adaptation in multi-objective particle swarm optimization*, IEEE Swarm Intelligence Symposium, 2006.

[63] S. Ho, S. Yang, G. Ni, E. Lo, H. Wong, "A particle swarm optimization-based method for multi-objective design optimizations", *IEEE Transactions on Magnetics*, vol. 41, no. 5, pp. 1756-1759, 2005.

[64] C. Raquel, P. Naval, *an effecive use of crowding distance in multi-objective particle swarm optimization*, In Proceedings of the GECCO, 2005.