

# Deformable-Medium Affordances for Interacting with Multi-Robot Systems

by

Matteo Diana

Submitted to the Department of Information Engineering  
in partial fulfillment of the requirements for the degree of

Master of Science in Robotics and Control Systems Engineering

at the

UNIVERSITÀ DI PISA

June, 14 2013

© Matteo Diana, MMXIII. All rights reserved.

Author .....

Department of Information Engineering

May 18, 2013

Certified by .....

Lucia Pallottino

Professor

Thesis Supervisor

Certified by .....

Magnus Egerstedt

Professor

Thesis Supervisor

Certified by .....

Andrea Caiti

Professor

Thesis Supervisor



# **Deformable-Medium Affordances for Interacting with Multi-Robot Systems**

by

Matteo Diana

Submitted to the Department of Information Engineering  
on May 18, 2013, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Robotics and Control Systems Engineering

## **Abstract**

In this thesis, I address the issue of human-swarm interactions by proposing a new set of affordances that make a multi-robot system amenable to human control. An affordance, as defined by Gibson [11], is a relation between an object and a user, where the object explicitly allows the user to perform a particular action. The identified affordances when controlling a swarm, include stretching the swarm, molding it into a particular shape, splitting and merging sub-swarms, and mixing of different swarms. The contribution beyond the formulation of these affordances is the coupling of an image recognition framework identified by an effective deformable-medium control interface, and the accompanying algorithms needed to identify the appropriate inputs, and then turn those into decentralized control laws for the individual robots. As result, the developed human-swarm interaction methodology is applied to a team of mobile robots.

Thesis Supervisor: Lucia Pallottino  
Title: Professor

Thesis Supervisor: Magnus Egerstedt  
Title: Professor

Thesis Supervisor: Andrea Caiti  
Title: Professor



## **Acknowledgments**

First I would like to thank my advisors Magnus Egerstedt and Lucia Pallottino for their support and helpful guidance. Their knowledge and advice have helped me to keep on track and work at a smooth pace. My gratitude goes out as well to Prof. Anthony Yezzi for his advices and support. Careful revisions of my essays have honed my research and organizational skills. I am indebted to my school colleagues who have helped make my learning an enjoyable and stimulating experience. I wish to thank my family and my close friends, whose have given me the motivation to aspire and realize this achievement. Without all these elements, such an achievement could not have been accomplished.



# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
<b>2</b>	<b>Swarms: How to Deal With Them?</b>	<b>17</b>
2.1	The Key is Multi-Agent . . . . .	17
2.2	Swarm Robotics . . . . .	19
2.3	Interaction Problems . . . . .	20
<b>3</b>	<b>The Algorithmic Framework</b>	<b>23</b>
3.1	Image Recognition Framework . . . . .	24
3.1.1	Creation of the Shapes Library . . . . .	25
3.1.2	Multisegmentation and Classification . . . . .	32
3.1.3	Outline of the Image Recognition Framework . . . . .	36
3.2	Distributed Swarm Control Framework . . . . .	37
3.2.1	Agent Dynamics . . . . .	39
3.2.2	The Control Laws Library . . . . .	40
3.3	Outline of the Algorithmic Framework . . . . .	45
<b>4</b>	<b>Simulations</b>	<b>47</b>
4.1	Discussion of Simulations . . . . .	53
<b>5</b>	<b>Experimental Results</b>	<b>57</b>
5.1	Khepera III . . . . .	59
5.2	Deformable-Medium . . . . .	59
5.3	Experimental Set Up . . . . .	60

5.4	Experimental Cases and Results . . . . .	61
5.5	Discussion of Experimental Results . . . . .	67
<b>6</b>	<b>Conclusions</b>	<b>69</b>



# List of Figures

1-1	Examples of affordances. . . . .	12
3-1	Structure of the algorithm. . . . .	24
3-2	Training Data: 10 2D binary shape models of the U-shape. . . . .	29
3-3	Alignment results of the Fig. 3-2 shape models. . . . .	29
3-4	Implicit shape priors: mean shape $\bar{\Phi}^{U-shape}$ . . . . .	32
3-5	A binary image of the clay with the zero levels set of the current $\Phi^{U-shape}$ superimposed to demonstrate the inner and outer regions used to compute the cost in the BMM. . . . .	34
3-6	Sections definition. . . . .	44
4-1	Some results for hundred mobile agents. . . . .	49
4-2	<i>Heterogeneous stretched-like</i> formation with reference orientation $\theta_{ref} = \frac{\pi}{4}$ . . . . .	50
4-3	<i>Heterogeneous U-like</i> formation with reference orientation $\theta_{ref} = \frac{2\pi}{3}$ . . . . .	51
4-4	<i>Heterogeneous S-like</i> formation with reference orientation $\theta_{ref} = \frac{2\pi}{5}$ . . . . .	51
4-5	Some split results for hundred mobile agents. . . . .	52
5-1	Khepera III. . . . .	58
5-2	Clay interface representing two shapes examples: an <i>U-like</i> shape and a Ball-shape. . . . .	59
5-3	Experimental set up. . . . .	61
5-4	<i>U-like</i> shape superimposed on the swarm: $\theta_{ref} = 0$ . . . . .	62
5-5	Trajectories of the swarm of Fig 5-4. . . . .	62
5-6	<i>U-like</i> shapes . . . . .	63

5-7	<i>U-like</i> shape superimposed on the extended swarm: $\theta_{ref} = 0$ . . . . .	65
5-8	Trajectories of the extended swarm of Fig 5-7. . . . .	65
5-9	<i>U-like</i> shapes: extended swarm case. . . . .	66

# Chapter 1

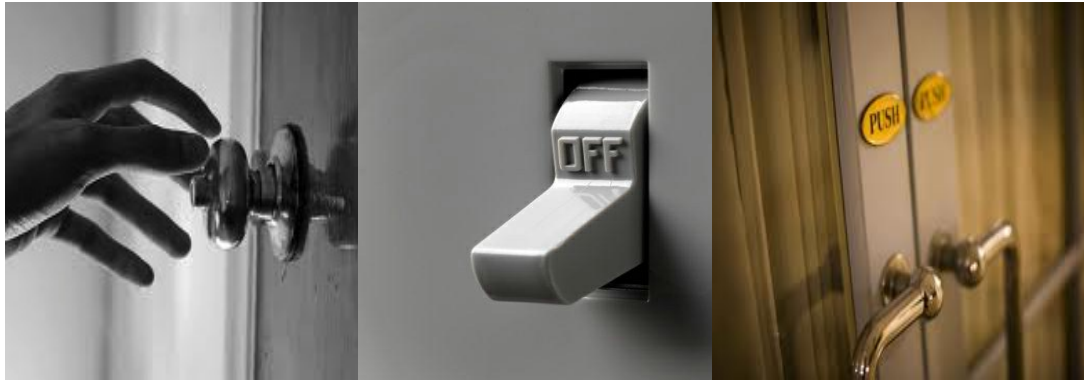
## Introduction

*“While human ingenuity may devise various inventions to the same ends, it will never devise anything more beautiful, nor more simple, nor more to the purpose than nature does, because in her inventions nothing is lacking and nothing is superfluous.”*

**Leonardo da Vinci**

The primary goal of this thesis is to employ current theoretical knowledge to achieve a human-swarm interaction methodology that allows an untrained operator to control a large collection of mobile robots. The question this thesis tried to answer is “Can a swarm be amenable to human control?” A few different approaches for interacting with swarms of mobile robots have been previously proposed. The main line of thought is to use so-called leader-based interactions, where the user interacts directly with a small subset of the agents, as is the case in [1], [2]. This is an effective strategy if the number of agents is relatively small. But, it becomes cumbersome as the number of agents increases, as was shown in [3]. Alternative approaches that have been proposed include induced flows across agents [4], boundary value control [5], or behavioral interactions [6], [7], [8]. A related question concerns the appropriate structure of user interfaces that ensures that sufficient situational awareness is provided and that the user is not overloaded with swarm-related inputs [9], [10].

The general way this relatively new problem has been addressed so far is starting from notions of the interaction dynamics, and then approaching the human-swarm interaction



(a) Knob - Rotate.

(b) Switch - Flip.

(c) Handle - Push.

Figure 1-1: Examples of affordances.

problem as one concerning how this interaction dynamics can be effectively manipulated. The approach taken in this thesis is the opposite, focusing on what constitutes effective *affordances* for interacting with large collections of mobile robots. An affordance, as defined by Gibson [11], is a relation between an object and a user, where the object allows the user to perform a particular action. Gibson's affordance implies complementarity of the user and the object. It is neither an objective nor a subjective property, and at the same time it is both. Affordances only make sense from a system point of view., e.g. system user-object. Affordances are invariants and are holistic, e.i. what we perceive when we look at objects are their affordances, not their dimensions and properties.

In Fig. 1-1 the affordances are rotate-able, flip-able, and push-able respectively. The objects in the environment are knobs, buttons and handles. The action of pushing is for handles rather than knobs, the constraint on handles allows it only to be pushed. Clearly, the definition of affordance is an intriguing, useful and too controversial concept to be expressed in only few words. An idea, more or less vague, of the meaning of the term affordance has been given since it is critical for this work. The purpose of investigate the appropriate definition of affordance goes beyond the scope of this thesis and many contributions to this topic have been witnessed in the years and examples of such contributions can be found in [12], [13].

It has been briefly described what it is assumed for affordance since this allowed to bridge the gap between the operator and the multi-robot control interface developed. This control interface brings into play another aspect of this thesis which relies on solving the issue of

controlling a multitude of agents. This problem can be part of a branch of robotics that goes under the category of bio-inspired robotics where models and algorithms try to capture useful behaviors from biology.

Animal aggregations are one of the most impressive visible patterns in biology. They often behave as a unit with properties that are not merely a sum of the individual behaviors. Such spatial distributions of individuals are almost universal across living organisms, from bacteria to higher vertebrates [14]. While the results of these aggregations are often visible, the individual-level mechanisms that are the cause of those brilliant results are often unknown or only partially understood. Underneath the emergent behavior its members could either work in synchrony e.g. fish, birds or not e.g. insects. The new functionalities that these behaviors bring include ability to build a nest or thermoregulate the hive e.g. bees, termites or as a group to perform a well-regulated density profile e.g. schooling fish. For instance, insects do not rely on sophisticated internal states, directed communication, global position and range information. Insect behavior is robust to environmental changes because very little processing occurs from when an insect perceives a new input and when it moves. Individuals in such groups interact only with their neighbors, but those may represent a vanishingly small fraction of the whole group.

Lots of questions may easily arise when one tries to understand such behaviors, i.e. what kind of features of spatial aggregations do confer advantages or disadvantages in the suitability of group members and how costs and benefits have been balanced during the evolution process both at the individual and group levels. Simply, the appeal of studying these systems relies on the failure of an external human observer to understand the rules that govern such simple individuals that emerge in a complex group behavior. Note that the term complex has the function of descriptor for global features of a self-organizing system. Self-organization produces complex behavior compared to the complexity of the individual agents producing the global organization. For example, in social insects, the complexity of an individual is clearly not sufficient to explain the organization, robustness and flexibility exhibited in an insect colony. Hence, these systems are potentially the source to solve real-world problems that make researchers to scratch their heads.

While the applications of swarm intelligence have propagated in various fields of engineer-

ing and computer science into business, telecommunications, finance, social psychology, etc. robotics-based applications are the focus of this research. Mathematical models of social behaviors such as flocking in birds, schooling in fish provide useful tools to exploit, in artificial environments, similar mechanisms of those observed in biology and nature in general. One of the most interesting features that clearly emerge from biology is how multiple simple individuals could perform complex tasks once put together. Such systems, where each individual follows a simple set of rules, provide automatically positive features to the whole group such as scalability, decentralization, parallelism, exploitation of peer-to-peer or via the environment local communication mechanisms among relatively simple agents, and also self-organization, reliability and efficiency. The abilities of such natural systems appear to transcend the abilities of the individual components which are ruled by nothing more than a simple set of low-level interactions.

However, even if it would be possible to provide a model which is only slightly different from the natural one, still it would not solve all problems. In an idealized situation, the robotic system should precisely duplicate the system behavior predicted theoretically. However, the constraints and uncertainties of real-world implementation make such an expectation unreasonable. Therefore, it is necessary to limit the scope of the experimental system and focus on several key elements of the case of study.

Provided the motivations and bases of the research, an overview is presented to provide the reader with a broad perspective of each chapter in relation to the complete work. Chapter 2 exposes some of the key problems one encounters when passing from natural swarms to artificial ones. Special attention is given to the interaction problems that naturally arise. In Chapter 3, the proposed algorithmic framework is detailed. Image recognition and distributed control laws will be extensively treated. Computer simulations in Chapter 4 demonstrate many of the theoretical propositions presented in Chapter 3. The simulations also allow to study the validity of the proposed algorithm and the results will be applied to large-scale swarms that involve as many as hundred individual agents. Simulations are analyzed and discussed in optic of the effectiveness of the developed algorithm. In Chapter 5, the details of robot design, experimental setup, and experimental results are provided. An analysis and discussion of the experimental results is then given. Chapter 6 contains

general conclusions, lessons, and directions for future research.





# Chapter 2

## Swarms: How to Deal With Them?

*“Every ant knows the formula of its ant-hill,  
every bee knows the formula of its beehive.  
They know it in their own way, not in our way.  
Only humankind does not know its own formula.”*

**Fyodor Dostoyevsky**

### 2.1 The Key is Multi-Agent

Pursuing the positive features observed in nature, a relative new field of research involve multi-agent systems, where, above all, each agent need to be simple and basing its actions just on local information. The source of such scheme is the pioneering work by Reynolds [15], who simulated a flock of birds in flight (using a behavioral model based on few simple rules and only local interactions). This field is heavily influenced by natural systems and other pioneering robotics researchers such as Fukuda et al. [16], Beni [17], Brooks [18], have contributed to build the common sense for swarm in artificial environments. Since then the field has witnessed many developments applying different approaches for swarm coordination, navigation and control just to cite a few.

Multi-agent or multi-robot systems can solve parallel subproblems reducing the time needed to perform the task of the entire system and reducing the computational cost, since the subproblems that each agent solves are simpler than the main one. The absence of a single

decisional center provides robustness and reliability with respect to faults of the complex system and the goal is to obtain better performance. The design of a multi-agent system may lead to heterogeneous agents specialized and possibly simpler and less expensive than a single complex agent.

However, multi-agent systems give a rise to new problems, i.e. coordination of a high number of agents, manage heterogeneous agents in a uniform fashion, succeed in tasks using not centralized, but local information and provide appropriate protocols for the communication. This architecture forces the methods and algorithms used to be scalable and deal only on local information. Handling heterogeneous groups of agents requires an appropriate abstraction of the problem to provide a correct formalization of the latter and effective resolution techniques. Heterogeneous robots may have different type of sensors and the geometry modeling of those sensors could be complex.

Many solutions of this problem have been proposed, but one of them is particularly useful for the following study: instead of trying to model the sensors geometry, one can please of modeling just “who senses who”. In this way, one loses much of the geometry information of the problem, but on the other hand, is allowed to design easier control laws based on the topology of the network built from this framework. These assumptions give a rise to use graph theory to handle multi-agent systems and the stability of the latter is shown to rely on the connectivity properties of the graph that represents agent interconnections, in terms of not only asymptotic convergence, but also robustness with respect to fault agents. Given the concept of a multi-agent system, if now things get more complex assuming that one has to deal with large scale groups of robots, the features that have been just discussed become extremely important regarding the efficiency, stability and success of the tasks assigned to such systems. Considering that lots of models have been developed starting from nature observations, it is straightforward that also the terminology have been borrowed from biology.

## 2.2 Swarm Robotics

Before proceeding, since exist many different approaches used in multi-robot systems such as distributed robotics or collective robotics, it is necessary to clarify the meaning of swarm robotics. This concern was first explicitly stated in [19] and a definition was provided as:

**Definition 1.** *Swarm robotics is the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among the agents and between the agent and the environment.*

This definition is what most closely tries to incorporate the bases of an artificial swarm. A set of desirable features for human-swarm interaction is identified based on the principles of swarm robotics. The system level operation of a swarm robotic system should exhibit three functional properties that are observed in natural swarms and remain as desirable properties of multi-robot systems [20].

- *Robustness.* A swarm robotic system should be able to operate despite disturbances from the environment or the malfunction of its individuals. First, swarms are intrinsically redundant systems; the loss of an individual can be immediately compensated by another one. Second, coordination is decentralized and therefore the destruction of a particular part of the swarm is unlikely to stop its operation. Third, the individuals that make up the swarm are relatively simple, making them less prone to failure. Fourth, sensing is distributed, hence the system is robust against the local perturbations in the environment.
- *Flexibility.* The individuals of a swarm should be able to coordinate their behaviors to afford tasks of different nature.
- *Scalability.* The swarm should be able to operate under a wide range of group sizes and support large number of individuals without impacting performance considerably. That is, the coordination mechanisms and strategies to be developed for swarm robotic systems should ensure the operability of the swarm under varying swarm sizes.

The robotic platform developed tries to include all these properties without losing the appropriate focus onto the goals of this thesis.

## 2.3 Interaction Problems

*Imagine that you are surrounded by a million robot mosquitos and you have a single “joystick” that you can use for interacting with the swarm. How should this interaction be structured? Which tasks will need to be executed by the operator rather than by the swarm?*

Recently, the way to structure interactions between human and a team of robots behaving as a swarm has attracted significant attention. Research is motivated by recent advances in technology, which provide new low-cost devices and computer science, that act as source of new theoretical knowledge. This topic has not a clear classification on its own, but mainly lies between Biology and Robotics instead. Controlling such Bio-Inspired systems is challenging due to the limitations of each individual robot and the large number of robots that have to be coordinated to succeed in the desired task.

One of the factors limiting the use of robot swarms in real-life environment is the lack of appropriate method for humans to interact with the swarm. Not all systems classes allow to clearly understand how to control them straight away, e.g. when controlling a rear-steering bike. The human intuition can miserably fail when faced to the control of some types of systems where is less explicit what the effect is when translating inputs into motions. Hence, a solution could be to abstract out the main features of that system and provide them in an understandable way to the operator, i.e. a sort of “translation” of the system parameters to something simpler and more intuitive for the operator. Though, this needs considerable advances in analytical tools and, in general, it has to be shown yet if this is even possible.

The majority of the studies in swarm robotics rely on algorithms based on autonomous operations of the robots, where the latter are often modeled starting from an insect, a bird, a fish, etc. For example, it is uncommon for humans to collaborate, drive or interact with insect swarms and hence the benefits of such an arrangement in swarm robotics have been

explored only in an infinitesimal part.

Lots of problems have to be investigated yet in this area, from the operator point of view, i.e. What kind of abstraction of the swarm should be used? What parameters play a key role to decide if an abstraction is better than one other? Or again what kind of medium should be used to communicate with the (whole) swarm? The aim of such abstractions is to being used by the operator to achieve the desired task, and they must provide the operator with sufficient information about the inner behavior of the swarm. Other problems arise from the swarm point of view, i.e. How the swarm can use its decentralized sensors to best perceive inputs from the human operator? An appropriate human interaction can benefit a swarm of robots to achieve goals more efficiently.

In this work, a human swarm interaction architecture is then proposed that has all of these desirable features. Despite all the efforts put in the research publications presented each year, swarm robotics has not yet been able to go far beyond laboratory research. While experiments allow researchers to study the system behavior in real environments, both to set up and the execution are costly and time-consuming. These downsides constrain the study of real swarm systems to the results obtained with just small size groups.

Models of swarm systems may help to identify the parameters that influence the individual and the team performance, and in general to gain useful insights into the system design. The approach taken in this work allowed to capture a level of abstraction of the swarm that turned out in an effective human-swarm interface. A deformable-medium has been used as “joystick” to efficaciously control the group of robots in its spatial distribution regardless of the group cardinality.



# Chapter 3

## The Algorithmic Framework

*“The formulation of the problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill.”*

**Albert Einstein**

The objective of the proposed algorithm is to provide the operator with an interface for interacting with a multi-robot system. By interacting is meant being able to modify the latter in its spatial distribution so that particular formations can be achieved. For interacting with multi-robot systems, rather than instrumenting the deformable-medium interface itself (e.g. clay), the proposed framework is based on a combination of computer vision algorithms for monitoring the clay interface, and distributed robotics algorithms for mapping the clay shape onto the robotic platforms.

The algorithmic framework relies onto two main frameworks – an image recognition framework and a distributed swarm control framework. Its structure is illustrated in Fig. 3-1. The image recognition framework (described in Section 3.1), consists of a fast segmentation-inspired algorithm that classifies each observed clay shape in the image to a shape class in the precomputed Shapes Library (SL). The shapes library contains all the allowed spatial distributions that the swarm can assume. Once the classification has been performed, the distributed swarm control framework (described in Section 3.2) must map the output from the image recognition process onto executable control laws. These control laws must rely solely on locally available information since, otherwise, the scalability of the human-swarm

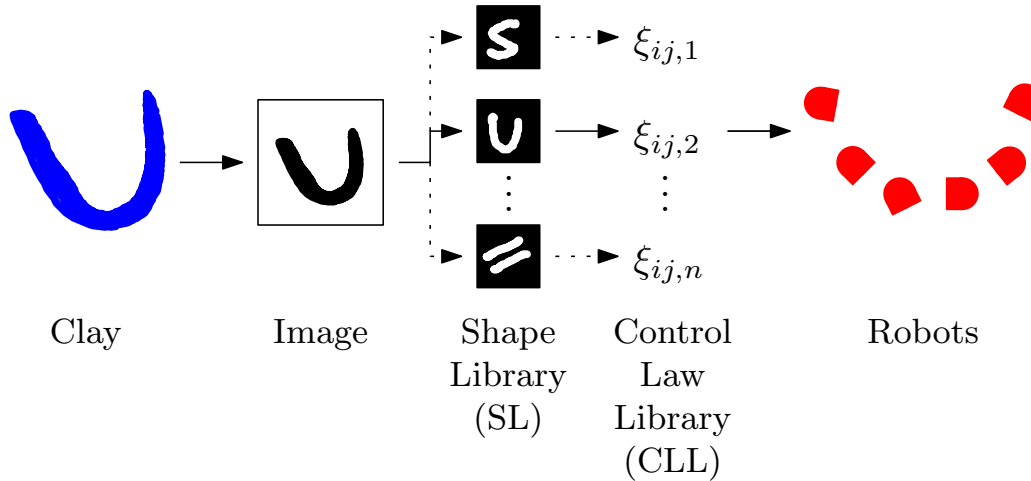


Figure 3-1: Structure of the algorithm.

interaction algorithm would suffer. As such, control laws will be developed such that the entire swarm can be controlled in its spatial distribution in a distributed and decentralized fashion, without being affected by the swarm size. The set of control laws forms a pre-computed Control Laws Library (CLL). The libraries just cited (the SL and CLL), are the result of off-line computations and between their elements there is a one-to-one correspondence, meaning that for each shape in the SL there exists a control law in the CLL.

The way a new shape is presented to be recognized is by modifying the clay, using the latter as robotic control interface. Whenever a new shape is observed the images recognition algorithm provides as result, in an on-line fashion, the appropriate control law for the swarm. That control law will eventually make the swarm to perform the desired shape.

### 3.1 Image Recognition Framework

In this section it is described the image recognition algorithm needed to classify the shape formed by the operator with the clay. The algorithm is divided into two main parts - the off- line part described in Subsection 3.1.1 and the on-line part described in Subsection 3.1.2. The objective of the off-line part was to extract the features of the boundary of a shape for each class and store these features in the SL. The objective of the on-line part was to classify the shape in an image of the clay based on the features stored in the SL.



### 3.1.1 Creation of the Shapes Library

The shapes library was created as detailed in the following. The first step was to decide how many shapes classes would have been part of the library, i.e. what spatial distributions it was desirable the swarm to perform. In the case of study, such shape classes were directly dependent on the affordances that were desirable when controlling the swarm. The identified affordances include stretching the swarm, molding it into a particular shape, splitting and merging sub-swarms, and mixing of different swarms. In terms of the image recognition framework, the latter affordances turned into *Line-Shape*, *S-Shape*, *U-Shape*, *Split-and-Line-Shape*, etc...

Next, for all shape classes an appropriate training set had to be available to extract the prominent shape variations within that class so as to reveal the inner details for further investigations. Note that any attempt to capture shape variations from an image training set must be preceded by an alignment process. The alignment process is required to remove any variations in shape due to pose differences. According to so, it has been applied first an alignment phase and then, it was created the SL.

#### Shape Alignment

The shape alignment is essential to exploit, integrate or compare different data. The alignment procedure in the field of Image Processing is called Image Registration and, in a simple case, given two images, one image is treated as the target image and the other is treated as a source image; the source image is transformed to match the target image.

There are different transformation models that could be applied to do so – in parametrized transformation fields a common choice is to use splines, if one needs invertible transformations with a smooth inverse the deformation fields that provide such property are the diffeomorphisms, in this framework rigid-body affine transformation models have been adopted. Another important consideration when performing image registration is about the similarity metric assumed. In this work, the registration quality within a shape class, was visually judged by overlapping all images in the training set in a pixel-wise fashion. The coherence of the aligned shapes was indicated by the increased sharpness of the summation

images meanwhile the blurriness indicated misalignment of the latter. More formal choices include using measures of the sum of squared distances (SSD), correlation coefficient, and mutual information. The acquisition noise can also play a key role in this decision. The last, but not the least important consideration to do is about the optimization procedure. Either continuous or discrete optimization is performed.

As it will be described later, two alignment methods have been tested – a coarsened alignment performed by hand and also a gradient-based optimization technique to jointly align the training images of each shape class. The principal solution adopted was the coarsened alignment performed by hand, but also the method proposed by [21] has been successfully implemented.

Under the assumption of 2D training binary images, similarity transformations are defined in order to jointly align different images representing shapes belonging to the same shape class (e.g. *U – shape*). Let  $m$  represent the number of shape classes. Whether method is employed, for each shape class one desire to segment<sup>1</sup>, a different images training set and alignment process is required. Let the training set  $T$  consist of  $n$  of such images. Specifically, let the training set  $T$  be defined as  $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n\}$ , where  $\mathbf{I}_i = (I_i^1, I_i^2, \dots, I_i^m)$  for  $i = 1, \dots, n$ . Each  $I_i^k$  for  $i = 1, \dots, n$  and  $k = 1, \dots, m$  is a binary image with values of one inside and zero outside the shape.

Both alignment methods rely on the pose parameters  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$  used to transform the  $n$  binary images to jointly align them. As previously said, the focus was on using body and scaling transformations to align these binary images to each other. In 2D, the pose parameter  $\mathbf{p} = [a \ b \ h \ \theta]^T$  with  $a, b, h$  and  $\theta$  corresponding to  $x$ –,  $y$ –translation, scale and rotation, respectively. The transformed image of  $I_i^k$  for  $i = 1, \dots, n$  and  $k = 1, \dots, m$ , based on the pose parameter  $\mathbf{p}_i$ , is denoted by  $\tilde{I}_i^k$ , and is defined as

$$\tilde{I}_i^k(\tilde{x}, \tilde{y}) = I_i^k(x, y),$$

---

<sup>1</sup>In computer vision, segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

where

$$\begin{aligned}
\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} &= T[\mathbf{p}_i] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} 1 & 0 & a_i \\ 0 & 1 & b_i \\ 0 & 0 & 1 \end{bmatrix}}_{M(a_i, b_i)} \underbrace{\begin{bmatrix} h_i & 0 & 0 \\ 0 & h_i & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{H(h_i)} \\
&\quad \times \underbrace{\begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R(\theta_i)} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{3.1}
\end{aligned}$$

The transformation matrix  $T[\mathbf{p}_i]$  is the product of three matrices: a translation matrix  $M(a_i, b_i)$ , a scaling matrix  $H(h_i)$ , and an in-plane rotation matrix  $R(\theta_i)$ . This transformation matrix  $T[\mathbf{p}_i]$  maps the coordinates  $(x, y) \in \mathbb{R}^2$  into coordinates  $(\tilde{x}, \tilde{y}) \in \mathbb{R}^2$ .

The solution of the alignment process is under-determined until the pose of one of the sample shapes is fixed and the rest of the shapes are aligned according to the pose of this reference shape. In the coarsened alignment performed by hand no further tools were required since one could have defined the pose parameter by looking only at each one of the original binary images and comparing the latter to the previously selected as the target one. A more effective and finer strategy to jointly align the binary images is to use a gradient descent to minimize the following energy functional

$$E_{align}^k = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\iint_{\Omega} (\tilde{I}^i - \tilde{I}^j)^2 dA}{\iint_{\Omega} (\tilde{I}^i + \tilde{I}^j)^2 dA} \tag{3.2}$$

where  $\Omega$  denotes the image domain and  $k$  is the number of the shape class. Different alignment processes were required for each shape class. Minimizing 3.2 is equivalent as to minimize the difference between any pair of binary images in the training collection within

the same shape class. The area normalization term in the denominator of 3.2 is again employed here to prevent all the binary images from shrinking to improve the cost function. The update equation for the pose parameter  $\mathbf{p}_i$  is given in terms of  $\nabla_{\mathbf{p}_i} E_{align}^k$ , which is the gradient of the energy functional  $E_{align}^k$  of the  $k$ th shape class taken with respect to the pose parameter  $\mathbf{p}_i$ . Specifically, the update equation results  $\mathbf{p}_i^{t+1} = \mathbf{p}_i^t - \alpha_{\mathbf{p}} \nabla_{\mathbf{p}_i} E_{align}^k$  where  $t$  denotes the iteration number and  $\alpha_{\mathbf{p}}$  denotes the step size in updating  $\mathbf{p}_i$ . The gradient descent method of above was then performed until convergence. This method jointly aligned the  $n$  example shapes in a shape class, and by doing this for each shape class one could have obtained the alignment of the complete set of training images.

The nature of the gradient descent approach described above allows only for infinitesimal updates of the pose parameters, thus giving rise fundamentally to – slow convergence properties and sensitivity to local minima. By using a multiresolution approach such weak points can be significantly reduced. The basic idea behind this is to use a coarsened representation of the training set to obtain good initial estimate of the pose parameters. Then, simply by refining these pose estimates by employing progressively increasing resolution images it is possible to mitigate the effects of the gradient descent convergence rate.

After one of such alignment procedures, the training shapes when compared against one another, within a shape class, shared roughly the same center, were all pointing roughly in the same direction, and were approximately equal in size. To illustrate this alignment process, Fig. 3-2 shows the raw training set for the class of U-shapes. Each of these examples depict an orthogonal view of an *U – like* shape which is one of the implemented shape classes. Notice that each performed shape was non regular and non uniform since the quality of the particular shape was directly dependent upon the manual skills of the operator. The *U – like* shape at the far left side of the figure is chosen to be fixed and its pose is used as a reference for the alignment process. The result of the alignment process applied to the case of the U-shape class is depict in Fig. 3-3.

The adopted alignment techniques are performed offline and for the case where a gradient descent is used to align the training images the convergence time was directly dependent on the quality of the employed images. When dealing with high quality images, a multiresolution approach is highly recommended to overcome the common unattractive features due

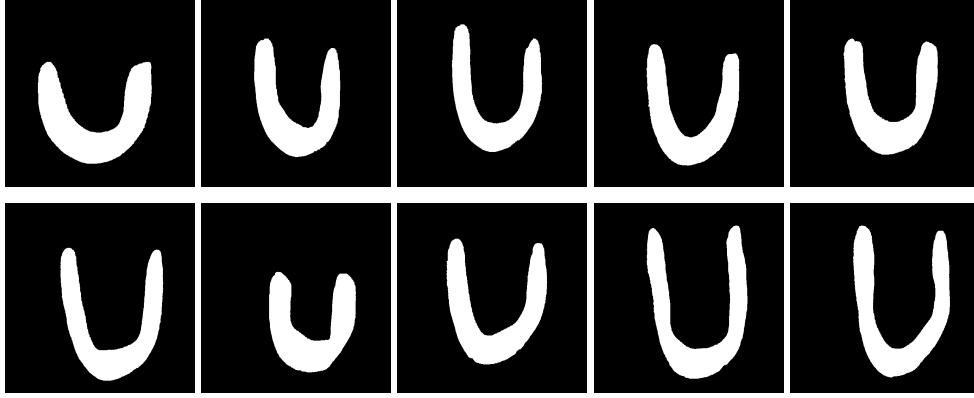


Figure 3-2: Training Data: 10 2D binary shape models of the U-shape.

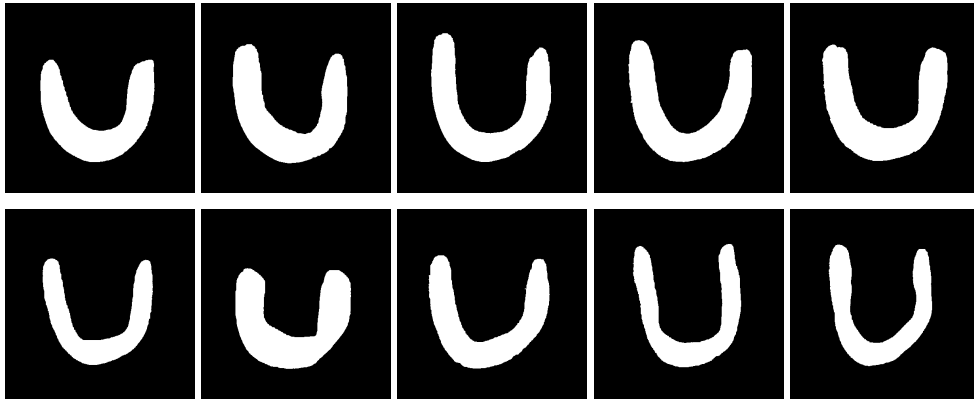


Figure 3-3: Alignment results of the Fig. 3-2 shape models.

to the gradient descent technique.

### Implicit Parametric Shape Models

Based on the level set methods proposed by Osher and Sethian [22], the technique followed is presented first in [21], where the shape representation have been casted in an Eulerian approach. Firstly, the signed distance function<sup>2</sup> has been chosen as representation for shape. Let  $k$  be the shape class. Specifically, for each shape class, the boundaries of each of the  $n$  aligned shapes in the database used for the alignment, were embedded as the zero level set of  $n$  separate signed distance functions  $\{\Psi_1^k, \Psi_2^k, \dots, \Psi_n^k\}$  with negative distances assigned to the inside of the object and positive distances assigned to the outside of the object. Then,  $\bar{\Phi}^k$ , the mean level set function of the shape database for the  $k$ th shape class has been

<sup>2</sup>The signed distance function  $\Psi(p)$  from an arbitrary point  $p$  to a known surface  $Z$  is the distance between  $p$  and the closest point  $z$  in  $Z$ , multiplied by 1 or  $-1$ , depending on which side of the surface  $p$  lies in [23]

computed, as the average of these  $n$  signed distance functions,  $\bar{\Phi}^k = (1/n)\sum \Psi_i^k$ . Next, within a shape class, the mean level set function was subtracted from each of the  $n$  signed distance functions. As result, the shape variabilities for each class were extracted from  $nm$  mean-offset functions:  $\tilde{\Psi}_i^k$  for  $i = 1, \dots, n$  and  $k = 1, \dots, m$ . These mean-offset functions were then used to obtain the variabilities within the  $m$  shape classes.

To capture the shape variabilities of a shape class,  $n$  column vectors were formed,  $\tilde{\psi}_i^k$ , consisting of  $N$  samples of each  $\tilde{\Psi}_i^k$ , where the sample locations had to be the same in each of the training images within the same shape class. Those samples were generated by a  $N = N_1 \times N_2$  rectangular grid of the training images and created in such a way that the columns of the image grid were sequentially stacked on top of one other to form one column. Next, the shape variability matrix for the  $k$ th shape class  $S^k$  was defined as

$$S^k = \begin{bmatrix} \tilde{\psi}_1^k & \tilde{\psi}_2^k & \dots & \tilde{\psi}_n^k \end{bmatrix} \in \mathbb{R}^{N \times n}.$$

An eigenvalue decomposition have been employed to factor  $(1/n)S^k(S^k)^T$  as

$$\frac{1}{n}S^k(S^k)^T = U^k \Sigma^k (U^k)^T$$

where  $U^k$  is a rectangular  $N \times n$  matrix whose columns represent the  $n$  principal variational modes or eigenshapes of the  $k$ th shape class, and  $\Sigma^k$  is an  $n \times n$  diagonal matrix whose diagonal elements, represent the corresponding non-zero eigenvalues. Each non-zero eigenvalue reflected the variance of shape variability associated with that eigenvalue's corresponding eigenshape.

The  $N$  elements of the  $i$ th column of  $U^k$ , denoted by  $U_i^k$ , were arranged back into the rectangular structure of dimension  $N_1 \times N_2$  by undoing the earlier stacking concatenation of the grid columns, to yeld the  $i$ th principal mode or eigenshape for the  $k$ th shape class denoted by  $\Phi_i^k$ .

In the end, this approach generated a maximum of  $n$  different eigenshapes  $\{\Phi_1^k, \Phi_2^k, \dots, \Phi_n^k\}$  for shape classes  $k = 1, \dots, m$ . Notice that in most cases, the computation of the eigenvectors and eigenvalues of the matrix  $(1/n)S^k(S^k)^T$  is computationally expensive since its dimensions are  $N \times N$ . In practice, the eigenvectors and eigenvalues of  $(1/n)S^k(S^k)^T$  can

be efficiently computed from a much smaller  $n \times n$  matrix  $W^k$  given by

$$W^k = \frac{1}{n}(S^k)^T S^k$$

It is straightforward to show that if  $d$  is an eigenvector of  $W^k$  with corresponding eigenvalue  $\lambda$ , then  $S^k d$  is an eigenvector of  $(1/n)S^k(S^k)^T$  with eigenvalue  $\lambda$  [24].

An useful design parameter was  $q < n$ , selected prior to segmentation, which is the number of modes to consider. The appropriate choice of this parameter was difficult and briefly,  $q$  should be bounded to a set of values that allow to capture the prominent shape variations present in the training set, avoiding to extend the set too much that the model begins to capture intricate details particular to certain training shapes. In all the examples addressed,  $q$  has been chosen empirically. In the end,  $m$  new level set functions have been introduced:

$$\begin{aligned} \Phi^1 [\mathbf{w}^1] &= \bar{\Phi}^1 + \sum_{i=1}^q w_i \Phi_i^1 \\ \Phi^2 [\mathbf{w}^2] &= \bar{\Phi}^2 + \sum_{i=1}^q w_i \Phi_i^2 \\ &\vdots \\ \Phi^m [\mathbf{w}^m] &= \bar{\Phi}^m + \sum_{i=1}^q w_i \Phi_i^m \end{aligned} \tag{3.3}$$

where  $\mathbf{w}^k = \{w_1^k, w_2^k, \dots, w_q^k\}$  for  $k = 1, 2, \dots, m$  are the weights for the  $q$  eigenshapes in each of the  $m$  new level set functions. The variances of these weights  $\{\sigma_1^l, \sigma_2^l, \dots, \sigma_q^l\}$  were given by the eigenvalues calculated earlier.

These newly constructed level set functions  $\{\Phi^1, \Phi^2, \dots, \Phi^m\}$  were proposed to be used as implicit representation of the  $m$  shape classes. It must be pointed out that by varying  $w^k$  it has been possible to try to match a clay shape performed by the operator. Specifically, the zero level set of  $\Phi^k$  described the boundaries of the  $k$ th shape class with that shape's variability directly linked to the variability of its level set function. Note that the shape variability allowed in this representation was restricted to the variability given by the  $q$  eigenshapes.

All the operations detailed so far were the result of off-line computations and provided the implicit parametric models needed for the segmentation of new images. These models formed the Shapes Library (SL). For the U-shape class, the implicit parametric shape model

resulted after the application of the level set methods described in [22, 21] is illustrated in Fig. 3-4b. Fig. 3-4a is its zero level set.

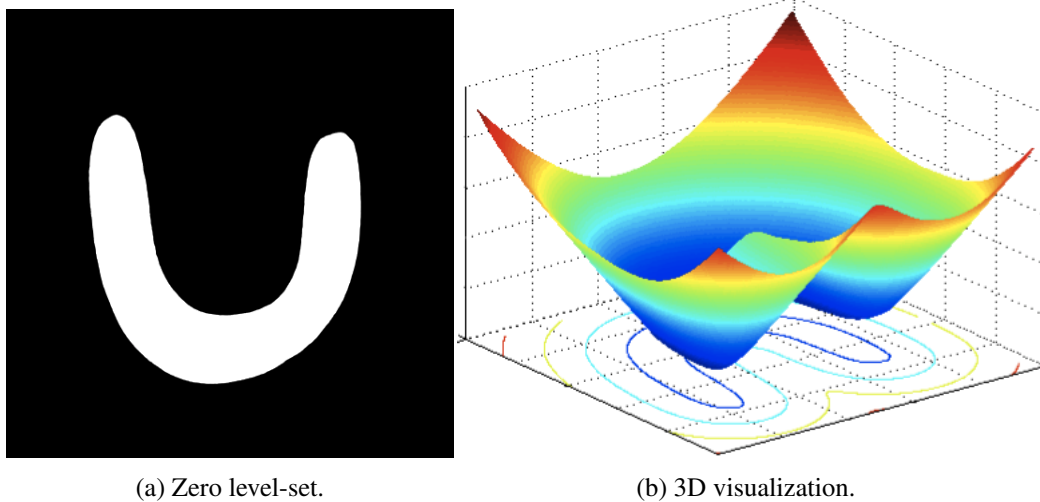


Figure 3-4: Implicit shape priors: mean shape  $\Phi^{U-shape}$ .

The storage of  $\Phi^{U-shape}$  in the SL allowed to complete the offline part of the image recognition algorithm for this particular shape class and provided the tools to classify if the shape of a molded piece of clay belonged to the class of U-shapes.

### 3.1.2 Multisegmentation and Classification

Following the lead of [21], it has been developed a region-based model to indirectly segment source shapes. In region-based segmentation models [25, 26, 27, 28] the evolution of the segmenting curve depends upon the pixel intensities within entire regions. That is, an image is seen as the composition of a finite number of regions and region-based models rely on regional statistics for segmentation. The statistics of entire regions are then used to direct the movement of the curve toward the boundaries of the image.

Specifically, in this section, it is presented a model for image segmentation and is described how it fits in the scope pursued. Region-based models generally are designed to derive the evolution equations for the curves used to segment the image. However, in this case, those models derive gradient descent equations used to optimize the shape parameters that indirectly describe the segmenting curve. In the following, a simple synthetic example



presents how the region-based segmentation model was incorporated into this model-based algorithm. The purpose of this example is to clarify the approach taken with respect to the segmentation procedure.

Assume that the domain of the observed image  $I$  is formed by two regions distinguishable by some region statistic (e.g. sample mean or variance). The goal is to segment this image via the curve  $\vec{C}$ , which in the followed framework, is represented by the zero level set of  $\Phi$ , i.e.

$$\vec{C} = \{(x, y) \in \mathbb{R}^2 : \Phi(x, y) = 0\}.$$

Note that  $\Phi$  is one of the level set functions derived in 3.3. Moreover, as result of this implicit parametric representation of  $\vec{C}$ , the regions inside and outside the curve, denoted, respectively, by  $R^u$  and  $R^v$ , are given by

$$R^u = \{(x, y) \in \mathbb{R}^2 : \Phi(x, y) < 0\}$$

$$R^v = \{(x, y) \in \mathbb{R}^2 : \Phi(x, y) > 0\}.$$

In the image recognition framework implemented, the parameters of  $\Phi[\mathbf{w}]$  were calculated to vary and hence segment the image  $I$ . The parameter,  $\mathbf{w}$  was obtained by minimizing a region-based energy functional that is constructed using various image statistics. Some useful image statistics, written in terms of  $\Phi[\mathbf{w}]$ , are

$$\begin{aligned} \text{area in } R^u: A_u &= \iint_{\Omega} H(-\Phi[\mathbf{w}]) dA \\ \text{area in } R^v: A_v &= \iint_{\Omega} H(\Phi[\mathbf{w}]) dA \\ \text{sum intensity in } R^u: S_u &= \iint_{\Omega} IH(-\Phi[\mathbf{w}]) dA \\ \text{sum intensity in } R^v: S_v &= \iint_{\Omega} IH(\Phi[\mathbf{w}]) dA \\ \text{average intensity in } R^u: \mu &= \frac{S_u}{A_u} \\ \text{average intensity in } R^v: \nu &= \frac{S_v}{A_v} \end{aligned}$$

where the Heaviside function  $H$  is given by

$$H(\Phi[\mathbf{w}]) = \begin{cases} 1, & \text{if } \Phi[\mathbf{w}] \geq 0 \\ 0, & \text{if } \Phi[\mathbf{w}] < 0. \end{cases}$$

Yezzi et al. in [28] proposed a pure region-based model to segment  $I$  using these region statistics and called Binary Mean Model (BMM).

Since the optimization of the energy functional is based on  $\mathbf{w}$  instead of  $\vec{C}$  this approach can be considered as a parameter optimization technique. It is proposed to evolve  $\vec{C}$  so as to maximize the distance between  $\mu$  and  $\nu$ . A natural cost functional they employed is to minimize the following:

$$E_{binary} = -\frac{1}{2}(\mu - \nu)^2 = -\frac{1}{2} \left( \frac{S_u}{A_u} - \frac{S_v}{A_v} \right)^2. \quad (3.4)$$

Graphically, the elements of this cost functional are illustrated for the U-shape class in Fig. 3-5. The interested reader could find more details in [21, 28] on the derivation of the

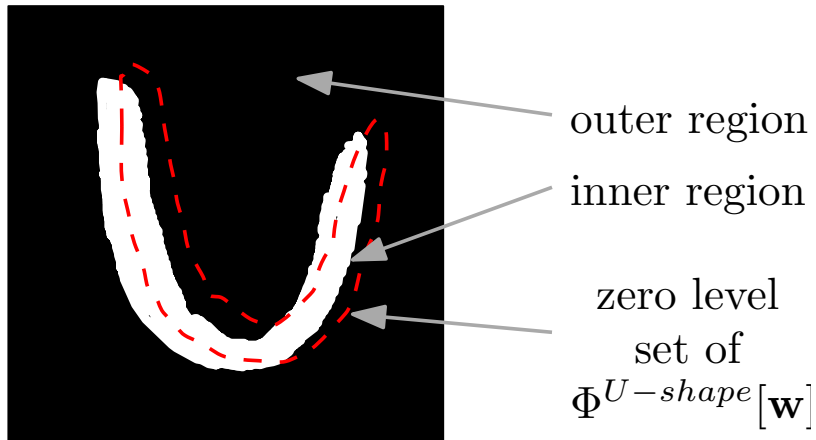


Figure 3-5: A binary image of the clay with the zero levels set of the current  $\Phi^{U-shape}$  superimposed to demonstrate the inner and outer regions used to compute the cost in the BMM.

update equations for the evolution of  $\vec{C}$ . Note that this framework does not require any particular cost functional, so one could easily fit different ones according to the specific case of study.

Starting from the Binary Mean Model the update equations obtained are given by

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha_{\mathbf{w}} \nabla_{\mathbf{w}} E_{binary}$$

where  $\alpha_{\mathbf{w}}$  is a positive step-size parameter, and  $\mathbf{w}^t$  denotes the value of  $\mathbf{w}$  at the  $t$ th iteration. The updated weights were then used to implicitly determine the updated location of the segmenting curve. It is important to notice that no special numerics is required in this technique as it does not involve any partial differential equations.

This results in fast and simple implementation of their methodology and is the model that has been used to provide the segmentation of source images. The parameter optimization technique previously detailed was embedded in the image recognition algorithm to provide such recognition.

### **Shape recognition**

The algorithm described above fits in the image recognition framework since the scope was to recognize each time what shape class (of the SL), the shape performed by the operator belonged to. Note that since the operator could change, different operators likely do not share the same manual skills when deforming the clay; therefore, all resulting shapes will be non regular and non uniform. Moreover, the segmentation process has no assumptions on what shape class of the SL (i.e. which  $\Phi^k[\mathbf{w}^k]$ ) should be used to segment the observed shape, so in a parallel fashion, the developed recognition algorithm tries to segment the new presented shape using each one of the models available in the SL.

Within each shape class, the segmentation process based on the minimization of the Binary Mean Model energy functional needed time to converge to its final segmented shape which was an unattractive feature in this case, since it was required to perform on-line shapes recognition. The proposed solution was to break the segmentation of the new shape before its convergence. This was made by letting the gradient descent optimization to compute only few updates of the zero level set of  $\Phi^k$ , the implicit parametric shape model of the curve  $\vec{C}$  where those updates resulted from the BMM energy functional. After those few updates the BMM energy functional has been evaluated for each of the implicit parametric

shape models available in the SL and the least value across all shape classes would have been reached only by the corresponding shape class of the new presented shape.

For example, suppose that the operator molds an *U-like* shape and that the precomputed SL consists of U-shape and Line-shape classes. The developed algorithm will try to segment the observed *U-like* shape using both  $\Phi^{U-shape}$  and  $\Phi^{Line-shape}$ . Since the models in the SL capture the characteristics of a particular class of shapes, when the segmentation starts using  $\Phi^{U-shape}$ , it will result in greater decrease in the BMM cost function than using  $\Phi^{Line-shape}$  for the segmentation.

The proposed algorithm handles pose differences in the observed images by matching the latter with the corresponding parameterized shape models after the classification phase. The method described above allowed us to perform on-line shapes classification using the models stored in SL. The developed framework allowed one to add as many shapes/control laws as desired if and only if the following held: the number of control laws was always equal to the number of shapes in the SL. This condition had to hold all times for the matching problem to be determined. So, adding a new element was a two-steps process which required to update the SL and CLL with the implicit parametric model of a new shape and a control law respectively.

In the end, the image recognition algorithm could have handled objects shapes that may differ from each other in terms of scale, orientation, or center location.

### 3.1.3 Outline of the Image Recognition Framework

In this section, a brief overview of the algorithmic framework is provided. The recognition algorithm can be divided into two phases – a training phase and a recognition phase. The training phase consists of shape alignment and parametric shape modeling (described in Section 3.1.1).

For each shape class, coarsened alignment performed by hand was employed to jointly align the given set of training shapes. Signed distance maps were generated to represent each of the shapes in the aligned database within each shape class. By applying PCA to this collection of distance maps, the quantities used to form the implicit parametric shape

representations derived in 3.3 were extracted. In the end, this phase created a Shapes Library (SL).

The next part of the image analysis algorithm, the recognition phase (described in Section 3.1.2), involved calculating the weighting parameters of the implicit shape representations to minimize the  $m$  segmentation functionals. These minimizations were performed as an iterative process using gradient descent. At each gradient step,  $\mathbf{w}^k$  were updated to generate new level sets  $\Phi^k[\mathbf{w}^k]$  with  $k = 1, 2, \dots, m$ . The segmenting curves  $\vec{C}^k$  were implicitly determined by these new level sets. Based on the new position and shape of each  $\vec{C}^k$ , the update rules for each  $\mathbf{w}^k$  were recalculated. During the recognition phase these steps have been performed in a parallel fashion for each shape class letting only few updates to be computed. Hence, the evaluation of the energy functionals allowed to find which shape class was the corresponding of the new segmented shape. By breaking the segmentation process before its convergence the algorithm has been faster enough to perform on-line shape recognition.

## 3.2 Distributed Swarm Control Framework

In this section it is described how, through local information, in a distributed and decentralized manner, it has been possible to control large-scale swarms. It is assumed that agents do not share a global reference system, they do not have an unique ID and they act in an asynchronous and independent fashion. Each one of the following control laws will be chosen after the result of the image recognition algorithm (described in Section 3.1). Case to case, the designated control law would have been effective until the eventual convergence of the swarm to the desired spatial distribution provided by the observed image. The distributed swarm control framework assumed to have a set of mobile agents that could be either homogeneous or heterogeneous and had shared goals. Each agent was not aware of the intents of its group mates, even though its actions were producing an advance to the goal of others. By these assumptions, the result was a collective emergent behavior where a simple set of rules provided a global emergent behavior. The robots were assumed to have a first order integrator dynamics.

Under these assumptions, a *plethora* of problems easily arise. Generally, using simple multiple agents from the control point of view, one has to face with problems such as limitations in the sensors, in the allowed computational load, in the actuation system and in the communication one. Also the algorithms have to be designed according to the structure of the group of robots, e.i. decentralized and distributed.

The distributed swarm control framework is made by a set of precomputed control laws which defined the Control Laws Library (CLL). Note that between the SL and the CLL there is a one-to-one correspondence. Once the image recognition framework classified the new source shape, with respect to those belonging to the shapes library, as consequence, also a match was defined with respect to the set of the control laws.

Before deriving the control laws though, it is necessary the introduction of some formal definitions. Let  $N$  be the number of agents, let  $s$  be the space dimension where the agents live.

**Definition 1.** *Let  $r_i(t) \in \mathbb{R}^s$  for  $i = 1, 2, \dots, N$  be the state of agent  $i$  at the time  $t$ . It is assumed that the interaction dynamics were ruled by pairwise interactions, i.e. if agents  $i$  and  $j$  were connected, then they shared relative state information. Since there were no distinctions between agents, in the sense of leader-follower networks, each agent was designed to maintain the desired distance and orientation with respect its neighbors.*

As it will be shown, the stability of the swarm is built around results on algebraic graph theory. Algebraic connectivity affects the performance and robustness properties of the overall system. The stability properties of the interconnected multi-robot system will be theoretically established by combining results from control theory, distributed robotics and algebraic graph theory.

**Definition 2.** *In a graph representation,*

- *the agents in the swarm are described by nodes  $V = \{v_1, v_2, \dots, v_N\}$ ,*
- *the connections between agents become edges  $E = \{(i, j) \in V \times V | i \sim j\}$ , ( $\sim$  denotes adjacency) and the cardinality of  $E$  represents the number of edges.*

By these definitions, neighbors of agent  $i$  belong to a set  $N(i) = \{j | (i, j) \in E\} \subseteq V \setminus i$

Agents within distances smaller than a certain positive amount are interacting through artificial "forces". In the following it is assumed to deal with networks whose underlying graphs are undirected. Consider two underlying graphs – the communication graph  $G_{comm} = (V, E_{comm})$  and the sensing graph  $G_{sens} = (V, E_{sens})$ . Let  $R_{comm} > R$  be the communication and sensing ranges, respectively. In the case of study, agents were subjected to sensing constraints which could have given rise to topology variations of  $G_{sens}$ . Such variations did not affect the control laws since it has been assumed that  $G_{comm}$  was static and remained connected all times (i.e. communication constraints always satisfied).

### 3.2.1 Agent Dynamics

To define the agents' dynamics, general energy-based definitions [2] were used, which allowed agents to achieve distance-based formation controls. Consider a group of  $N$  mobile agents moving on the plane ( $s = 2$ ), with dynamics expressed by single integrator,

$$\dot{r}_i = u_i, \quad i = 1, 2, \dots, N \quad (3.5)$$

where  $r_i = (x_i, y_i)^T$  is the position of agent  $i$  and  $u_i = (u_{ix}, u_{iy})^T$  is its input. Let  $R$  be the sensing range and  $\|r_{ij}\|$  be the distance between agents  $i$  and  $j$ .

**Definition 3.** For each edge  $j$  incident to agent  $i$  it is defined a nonnegative potential function  $\xi_{ij}$  dependent on the distance and orientation between  $i$  and  $j$  such that,

- $\xi_{ij}$  has an unique minimum,
- $\xi_{ij}$  is monotonically increasing near  $\|r_{ij}\| = R$ .

Such edge-tension energy can be defined as

$$\xi(r) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \xi_{ij}(r_i(t), r_j(t)) \quad (3.6)$$

which is the summation of the local contributes

$$\xi_{ij}(r_i(t), r_j(t)) = \begin{cases} \frac{1}{2}\{e_{ij}(\|r_{ij}\|)\}^2 & (v_i, v_j) \in E_{sens} \\ V > 0 & (v_i, v_j) \notin E_{sens} \end{cases}$$

where  $e_{ij} : \mathbb{R}^+ \rightarrow \mathbb{R}$  is a strictly increasing function. Let  $d_{ij}$  be parameter that expresses the desired distance and orientation of the pairwise. Then,  $e_{ij}(d_{ij}) = 0$ . Each agent tries to minimize the related parts of the edge-tension energy 3.6 through gradient descent:

$$u_i = - \sum_{j \in N_s(i)} \frac{\partial \xi_{ij}(r_i(t), r_j(t))^T}{\partial r_i}, \quad i = 1, 2, \dots, N \quad (3.7)$$

where  $N_s(i) = \{j | (i, j) \in E_{sens}\} \subseteq V \setminus \{i\}$  is the sensing neighboring set of agent  $i$ . The problem has been recasted in a more useful way based on choices made so far. By stacking the position vectors of Definition 1 it was derived a different dynamical system. This system had  $\bar{r}$  as its state, where  $\bar{r} = (B_{K_N} \otimes I) r$  is the stack vector of all relative positions between agents,  $B_{K_N}$  is the oriented incidence matrix of the complete communication graph with  $N$  vertices,  $K_N$  represents an arbitrary orientation,  $\otimes$  denotes the Kronecker matrix product,  $I$  is the identity matrix of appropriate dimension and  $r$  is the stack vector of agent positions. In this case the dynamics could be expressed

$$\dot{\bar{r}} = (B_{K_N} \otimes I_2)u, \quad (3.8)$$

where  $u$  is the stack vector of all inputs defined in 3.7. Since  $G_{comm}$  remained connected all times the control interconnections were fixed and time invariant.

### 3.2.2 The Control Laws Library

In the following it is shown how different edge-tension energies modify the swarm spatial distribution by means of the local control laws defined in 3.7. The developed control laws form the Control Laws Library (CLL). Under the assumption that the agents initial positions were not all coincident, proofs of convergence will be given.

Let  $\theta_{ij}$  be the angle under which agent  $i$  sees agent  $j$  and let  $d_{i,loc}$  be the distance reached



so far by agent  $i$  with respect its (local) initial position.

## Stretching

An example of  $e_{ij}$  that allowed the swarm distribution to be stretched in the preferred direction and orientation is

$$e_{st,ij} = \|r_{ij}\|^2 - (\cos(\theta_{ij} - \theta_{ref})\Delta + \sin(\theta_{ij} - \theta_{ref})\delta))^2 \quad (3.9)$$

where the quantities  $\Delta$  and  $\delta$  represent the stretching factors with respect to the reference orientation  $\theta_{ref}$ . If  $\Delta = \delta$  then the swarm formation is a circle, while when  $\Delta \neq \delta$  then the swarm is stretched into an ellipsoid, which is stretched most in the direction of the larger of these two gains.

**Theorem 1 (Stretching).** *Consider a system of  $N$  mobile agents with dynamics 3.8, each steered by control law 3.7 with local energy as in 3.9. Then the system approaches a configuration that minimizes all agent potentials and all pairwise agents assume the desired interagent positions and orientations.*

**Proof:** Let  $R_{comm} > R$  be the communication range. Consider the function  $\xi$  that is differentiable and continuous everywhere. The communication graph diameter cannot be larger than  $(N - 1)$  since the graph is connected. This implies that the largest distance between any two agents in the graph is smaller than  $(N - 1)R_{comm}$ . Moreover, it results  $\sum_{(i,j) \in V \times V} \|r_{ij}\| \leq N(N - 1)^2 R_{comm}/2$  and  $\bar{r}$  always evolves in a closed and bounded set. Likewise, the level sets of  $\xi$  define compact sets and are bounded.

From connectivity it is known that a path connecting nodes  $i$  and  $j$  has length at most  $(N - 1)$ . Let  $a > 0$ , from the properties of  $\xi$ ,  $\|r_{ij}\| \leq \xi_{ij}^{-1}(a(N - 1))$ . So, the set

$$\Omega = \left\{ (\bar{r}) \mid \|\bar{r}\| \leq \frac{N(N - 1)^2 R_{comm}}{2} \right\} \quad (3.10)$$

is compact. We follow showing that  $\Omega$  is nonincreasing and then establishing the invariant properties of  $\Omega$ . Considering that,  $\partial_{r_{ij}} \xi_{ij} = \partial_{r_i} \xi_{ij} = -\partial_{r_j} \xi_{ij}$  and taking the derivative of  $\xi$

it follows,

$$\begin{aligned}
\dot{\xi} &= \frac{1}{2} \sum_{i=1}^N \dot{\xi}_i = \frac{1}{2} 2 \sum_{i=1}^N r_i^T \sum_{j \in N_{s,i}} \partial_{ij} \xi_{ij} \\
&= \sum_{i=1}^N (\partial_{r_i} \xi_i)^T \dot{r}_i = - \sum_{i=1}^N (\partial_{r_i} \xi_i)^T (\partial_{r_i} \xi_i).
\end{aligned} \tag{3.11}$$

The last term in 3.11 can be alternatively rewritten as  $-\sum_{i=1}^N \|\partial_{r_i} \xi_i\|^2 \leq 0$  which is negative semidefinite and null whenever the argument of the sum is null  $\forall i$ . Under the particular choice made on the inter-agent potential function  $\xi_{ij}$ , let

$t = (\cos(\theta_{ij} - \theta_{ref})\Delta + \sin(\theta_{ij} - \theta_{ref})\delta)$ , then the zeros of  $\dot{\xi}$  are those satisfying

$$\sum_{j \in N_s(i)} e_{st,ij} (r_i - r_j) = 0, \quad \forall i = 1, \dots, N$$

so either  $r_i = r_j$  or  $e_{st,ij} = ((x_i - x_j)^2 - (y_i - y_j)^2) - t^2 = 0$ . The latter represent a circle centered in agent  $j$ 's position with the radius function of the desired distance and orientation of the pairwise to which the agent  $i$  should belong.

Let the set  $\gamma = \{r_i \mid -\sum_{i=1}^N \|\partial_{r_i} \xi_i\|^2 = 0\} \subset \Omega$ . Applying LaSalle's invariance principle from initial conditions in  $\Omega$  the solutions of the system converge to a subset of  $\gamma$ , more precisely of  $\{r_i \mid x_i, y_i \in \text{span}\{\gamma\}\}$ . If that, agents' relative velocities  $\dot{r}_{ij} = 0$  and the system dynamics becomes  $\dot{r} = -(B_{K_N} \otimes I_2)[\dots \partial_{r_{ij}} \xi_{ij} \dots]^T$ . So both  $\dot{x}_i, \dot{y}_i$  belong to the range of the oriented incidence matrix  $B$  of  $G_{comm}$ . For a connected communication graph,  $\text{range}(B_{K_N}) = \text{span}(\gamma)^\perp$ . Thus, in the invariant set within  $\gamma$ ,  $x_i, y_i \in \text{span}\{\gamma\}$  and this implies that  $\dot{x}_i, \dot{y}_i \in \text{span}\{\gamma\}$ . The latter result allows to conclude that we are faced to a contradiction unless  $\dot{x}_i, \dot{y}_i \in \text{span}\{\gamma\} \cap \text{span}\{\gamma\}^\perp \equiv \{0\}$ .

In the end, the control law 3.7 is zero at steady state implying  $\xi_{ij}$  is locally minimized. If  $\xi_{ij}$  is (locally) convex within the communication range, then the extremum is unique and the agents are stabilized to their desired positions and orientations. Collision avoidance is not ensured by the definition of  $\xi_{ij}$  so an appropriate algorithm that provides that has been implemented. ■

The existence and uniqueness of the solutions of 3.8 is provided by the boundness of 3.7.

If one of the two assumptions (connection property of  $G_{comm}$  and initial positions not all coincident), cannot be guaranteed, it could happen though, that  $r_{ij} \notin \Omega$  and by consequence the stability is not guaranteed.

## Bending

A series of different spatial distributions that represent curve shapes were obtained by other choices of  $e_{ij}$ , e.g. *U-shapes*, *S-shapes*, *L-shapes* rotated by the desired angles. One of such potentials is

$$e_{bend,ij} = \|r_{ij}\|^2 - (\cos(\theta_{ij} - \lambda)\Delta + \sin(\theta_{ij} - \lambda)\delta)^2 \quad (3.12)$$

where

$$\begin{cases} \lambda = (\theta_{ref} + \sigma) \\ \sigma = \pm f(\|d_{i,loc}\|) \frac{\pi}{c} \end{cases} \quad (3.13)$$

with  $c > 0$  is a constant and

$$f(\|d_{i,loc}\|) = \begin{cases} 0, & \|d_{i,loc}\| \in sec1 \\ 1, & \|d_{i,loc}\| \in sec2 \\ 2, & \|d_{i,loc}\| \in sec3 \\ 3, & \|d_{i,loc}\| \in sec4 \end{cases} \quad (3.14)$$

Let  $sec^* > 0$  be sections of progressively increasing distances as illustrated in Fig. 3-6. The effect of  $\lambda$  is to modify the orientation angle of each pairwise depending upon which section an agent belongs to.

The statement and proof in the case of bending energy are intentionally excluded since they follow the same fashion of Theorem 1. Notice that the only change in the proof of the theorem would be restricted to one of the solutions of the derivative of  $\xi$ . More precisely, the solution of  $e_{bend,ij} = 0$  is a circle of a different radius since  $t = (\cos(\theta_{ij} - \lambda)\Delta + \sin(\theta_{ij} - \lambda)\delta)$ .

## Splitting and Organizing

The purpose of the following was to provide the (decentralized and distributed) splitting capability to the swarm. Then, the subgroups could have been arranged in different spatial distributions using one of the edge-tension energies described above. Once the swarm was

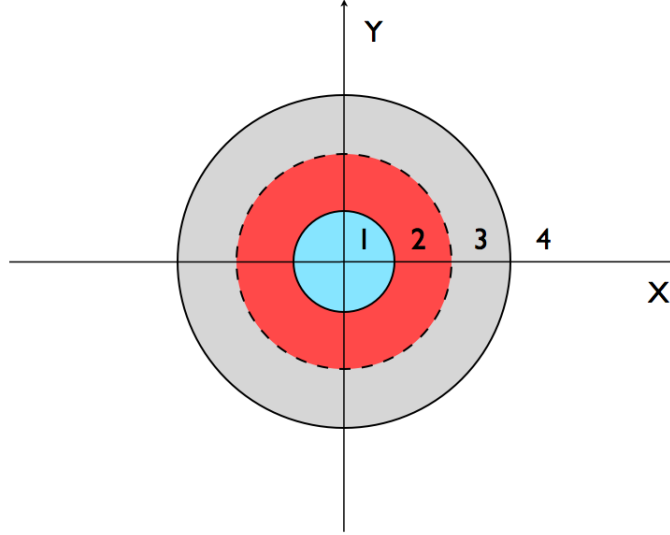


Figure 3-6: Sections definition.

separated, it could have happened that  $G_{sens}$  would have eventually lost the connection property. Nevertheless, it was still possible to ensure convergence of the system through the control law of the form in 3.7 since  $G_{comm}$  remained connected. The local contributes to the edge-tension energy in 3.6 had to be slightly modified to capture these features

$$\xi_{split,ij} = \begin{cases} \pm \frac{1}{2} r_i^2 k \text{sign}[\cos(\rho) \sin(\rho)], & \|d_{i,loc}\| \leq sec1 \\ \xi_{ij}, & otherwise \end{cases} \quad (3.15)$$

where  $\rho = \theta_{ref} - \frac{\pi}{2}$  and  $k > 0$  is a design parameter. This particular choice of  $\rho$  let the swarm to be split in the perpendicular direction with respect to the desired orientation  $\theta_{ref}$ , to assume.

As in the previous case, the statement and the proof of the splitting and organizing energy follow the same fashion of Theorem 1. Notice that no switching topologies were involved in this case since the communication graph remained connected.

## Merging

For the sake of completeness, it has been developed an appropriate strategy to allow the swarm to rejoin after it was split or to come back to its initial spatial distribution. The basic idea under the merging capability relies on the possibility to drive all agents to locally agree

on a value of interest through a consensus problem. Provided that each agent remembers its (local) initial position, by applying a self edge-tension energy such as

$$\xi_{i,loc} = \frac{1}{2}(\|d_{i,loc}\| - p)^2$$

where  $p \geq 0$  is the desired distance on which agree, all agents would eventually have returned to their initial positions. Even in this case, the statement and the proof of the merging energy follow the same fashion of Theorem 1.

### 3.3 Outline of the Algorithmic Framework

In this section, it is provided a brief overview of the proposed algorithmic framework. The proposed algorithm can be divided into two phases – an image recognition phase and a swarm control phase. Whenever the operator presented a new shape (part of the shapes library) by modifying the clay-based controller, the image recognition algorithm analyzed the observed image and classified it. As result, starting from the output of the recognition phase the corresponding control law (part of the control laws library), could have been selected. Then, this information has been sent to a random agent, which started to spread the signal to its neighbors. Thank to the properties of  $G_{comm}$ , that agent was connected with all its group mates and, by consequence, the whole swarm would have been subjected to the same control law. In the end, the control laws (described in 3.2.2) would have eventually driven the swarm to assume the desired spatial distribution.

A notable remark is that both frameworks did not share any special connection, so any change in one did not affect the other. This feature is highly desirable in such a problem since allows one to modify, redefine or create new frameworks and substitute the proposed ones in a simple, fast and functional manner.



# Chapter 4

## Simulations

*“Nothing happens until something moves.”*

**Albert Einstein**

Computer simulation is highly common within the swarm intelligence research community. The reason why simulations are the most common method of researching such systems is because they are way simpler in terms of implementation and time needed to run with respect to the real robots case, especially for a large number of agents. In the case of study, the overall aim was to test the effectiveness of the proposed algorithm and to investigate the predictions of theoretical modeling, to enhance understanding of self-organization with a collision avoidance behavior included.

In cases where theoretical results are either incomplete either absent, simulations are exceptionally important, as they advance the knowledge of swarm-intelligent systems where mathematical models are yet undeveloped.

The stability results of Chapter 3 have been verified under MATLAB environment. A set of four shape classes has been employed in the simulations to test the image recognition framework. Specifically, the shape classes were: *Ball-like*, *Stretch-like*, *U-like* and *S-like* shapes.

Regarding the distribute swarm control framework a larger set was employed. Three main categories of control laws were developed allowing a group of mobile agents to be stretched, bended and split in the desired direction and orientation. Combining the cate-

gories of above more possibilities emerged so that the swarm could have been split and stretched or bended to achieve various shapes, e.g. *S-like*, *U-like*, *split-and-stretched-like*, *split-and-U-like*, etc. Another separate category of control laws provided the merging capability whenever one desired to merge the swarm after it was split or to bring it back to its initial spatial distribution. Finally, the multi-robot system could have been either homogeneous either heterogeneous.

In the end, summarizing all the possibilities, were considered 3 different bending configurations and 1 stretched configuration. This led to 4 main configurations which could have been both mixed regarding to the homogeneous/heterogeneous property and again split. Considering all cases, up to 16 control laws were available and extensively tested.

Simulations followed the algorithm flow as depict in Fig. 3-1. Firstly, a synthetic image is processed during the image classification phase<sup>1</sup> and then, once the right shape class is identified, the multi-robot system is subjected to the appropriate control law<sup>2</sup>. Synthetic images were produced to verify the classification phase and according to what detailed in Section 3.1 results showed the effectiveness of the proposed methodology. Each time a synthetic shape was presented, as result, the recognition algorithm computed the corresponding control law provided by the right match between the SL and the CLL. The recognition algorithm required less than 3 seconds to observe and recognize the presented image on a PC equipped with a 2 Ghz Intel Core 2 Duo CPU and 4 GB 1067 Mhz DDR3 RAM memory.

Subsequently, the control laws were tested on a group of robots consisting of hundred mobile agents that were assumed to be dimensionless with identical first order integrator dynamics. Initial positions were generated by an uniform distribution in an area of  $10 \times 10$  units, centered at the origin. Simulations verify that the system converged to invariant sets that corresponded to the desired tight formations. Each time, the shape of the formations which the swarm converged to was determined by the energy functions described in Section 3.2.

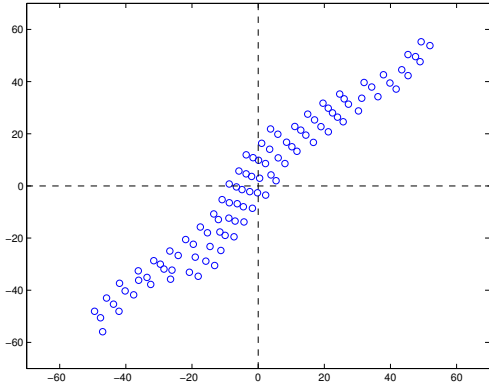
For sake of completeness the convergence time has also to be taken into account. For the

---

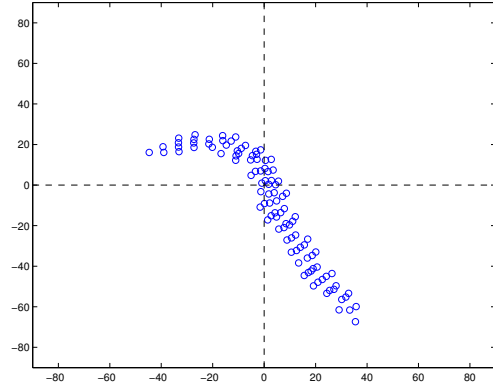
<sup>1</sup>The technique described in 3.1

<sup>2</sup>One of the control laws described in 3.2

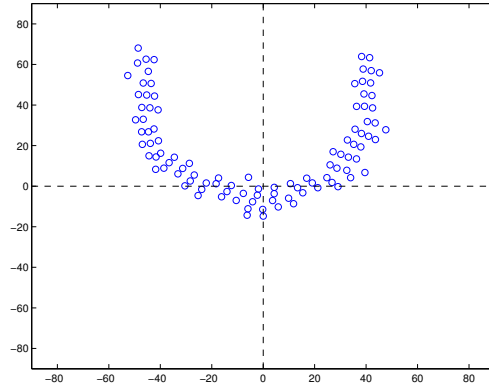




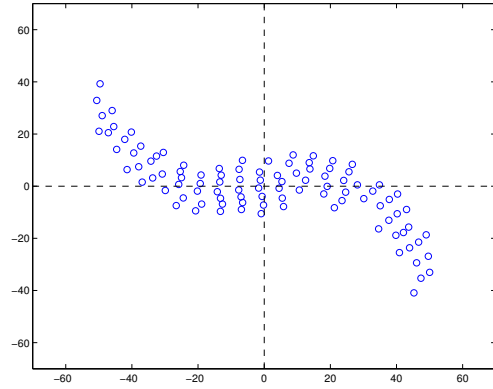
(a) *Line-like* formation: orientation  $\theta_{ref} = \pi/4$ .



(b) *Corner-like* formation: orientation  $\theta_{ref} = \frac{5\pi}{8}$ .



(c) *U-like* formation: orientation  $\theta_{ref} = 0$ .



(d) *S-like* formation: orientation  $\theta_{ref} = 0$ .

Figure 4-1: Some results for hundred mobile agents.

same PC as detailed previously the convergence times lie between 100 to 150 seconds. The greater bound was reached when performing the more complex shapes.

Extended large-scale simulations have been performed and some results are visible in Fig. 4-1. These results are obtained by the following parameters  $\Delta = 8$ ,  $\delta = 0.5$ , *sensing range*  $R = 6$ ,  $0 < sec0 \leq 10$ ,  $10 < sec1 \leq 20$ ,  $20 < sec2 \leq 33.3$ ,  $33.3 < sec3 \leq 50$  and  $sec4 > 50$ . Within each simulation the parameters are kept constant to the initialized values. Maintaining all the simulation parameters fixed throughout independent simulations, as the parameters  $\Delta$  and  $\delta$  varied, different stretched spatial distributions were obtained. The stretching direction was determined by the greater between these two gains. As the ratio between  $\Delta$  and  $\delta$  increased, the stretching of the robots formation also increased. Similarly, as the ratio was converging to the unit, the spatial distribution reached was non-stretched,

i.e. a circle. Note that, both gains were bounded within a set of values, the reason of this derived directly from the definition of the potential functions of Section 3.2.

A similar approach has been taken to test the sensitivity with respect to the variation of the other parameters. The way the variations in the sensing range affected the formation the swarm converged to is that smaller sensing ranges produced more compact and tight spatial distributions while bigger ones resulted in more loose and dilated distributions. One reason for this behavior is that since an agent with a bigger sensing range is able to “see” further, also has better chances to have more neighbors and that translates into bigger input. The latter meant also greater movements since the agents’ dynamics was a pure first order integrator.

The case of heterogeneous swarms has also been investigated and some fascinating behaviors emerged. The results of Figs. 4-2, 4-3 and 4-4 were obtained under the same set of

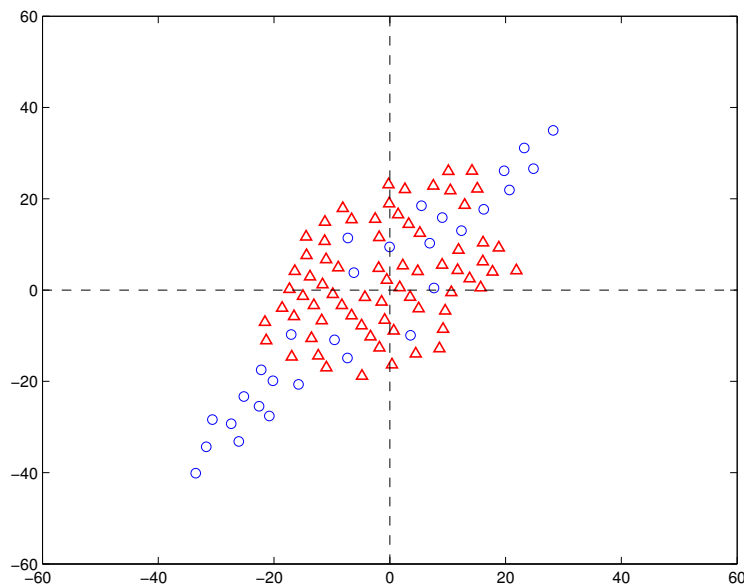


Figure 4-2: *Heterogeneous stretched-like* formation with reference orientation  $\theta_{ref} = \frac{\pi}{4}$ .

parameters as for the previous case. The only change was that since now there were two different types of robots, also two sensing ranges had to be defined. Specifically,  $R_{type1} = 6$  and  $R_{type2} = 4$ . Note that for the case of *heterogeneous stretched-like* formations, Fig. 4-2, there is not a notion of sections since the bending is not performed. Consider the case

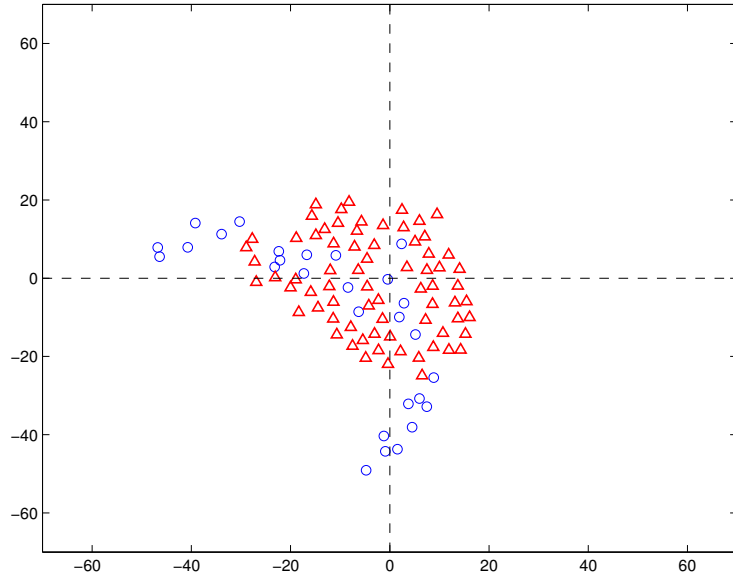


Figure 4-3: *Heterogeneous U-like* formation with reference orientation  $\theta_{ref} = \frac{2\pi}{3}$ .

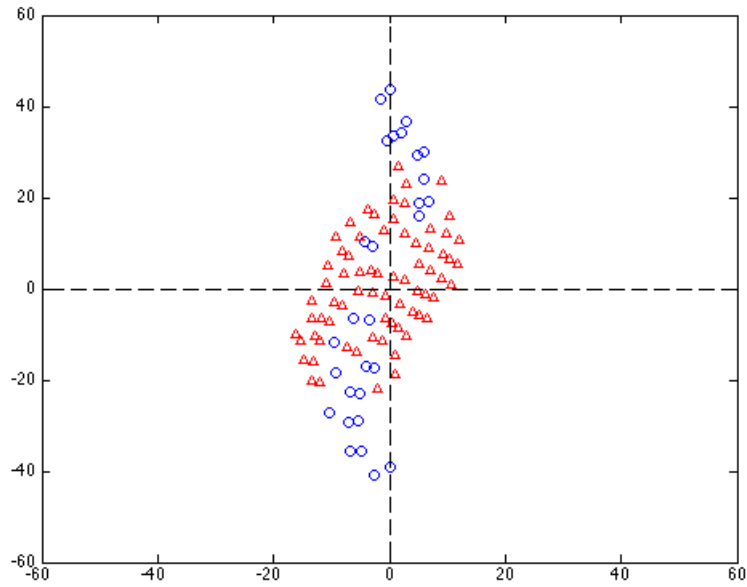
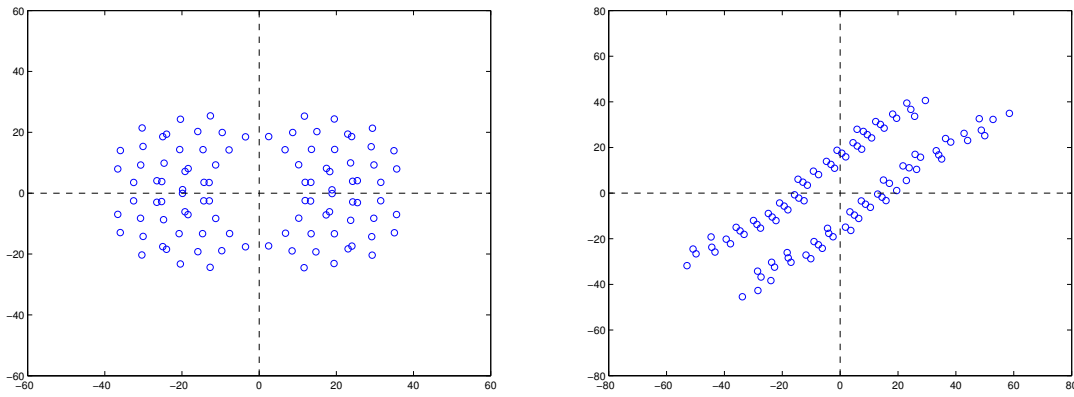


Figure 4-4: *Heterogeneous S-like* formation with reference orientation  $\theta_{ref} = \frac{2\pi}{5}$ .

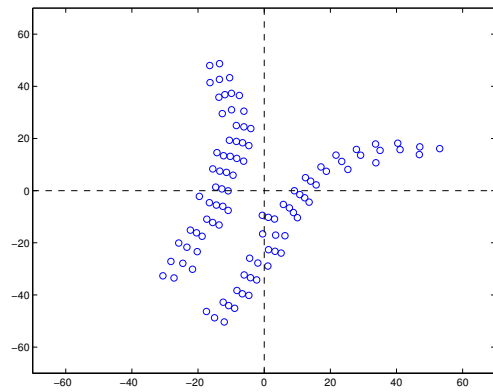
where there are two classes of robots  $r_{type1}$  and  $r_{type2}$ , with identical dynamics and control laws, but they differ in the sensing range  $R_{type1} > R_{type2}$ . In the figures,  $r_{type1}$  agents are the depicted as circles and  $r_{type2}$  ones as triangles. Using the same energy functions defined in

Section 3.2, the simulations in this case showed how  $r_{type1}$  agents tend to move themselves to the extremities of the formation. A possible explanation is given by the fact that since  $r_{type1}$  agents have a greater sensing range, they also have better chances to have a greater cardinality of the neighboring set with respect to  $r_{type2}$  agents. Hence, it is likely that the input for these agents is greater in magnitude.

Once these basic shapes were simulated effectively, a further step has been taken to test the distributed and decentralized splitting capability of the multi-robot system. Some results of split formations are depicted in Fig. 4-5. The parameters for those simulations were the



(a) *Split-circles-like* formation: orientation  $\theta_{ref} = 0$       (b) *Split-lines-like* formation: orientation  $\theta_{ref} = \frac{\pi}{4}$



(c) *Split-bends-like* formation: orientation  $\theta_{ref} = \frac{\pi}{3}$

Figure 4-5: Some split results for hundred mobile agents.

same as in the previous case with one class of robots, e.i.  $\Delta = 8$ ,  $\delta = 0.5$ , *sensing range*  $R = 6$ . Note that in the case of *split-bend-like* formations as in Fig. 4-5c, the sections were  $0 < sec0 \leq 10$ ,  $10 < sec1 \leq 20.8$ ,  $20.8 < sec2 \leq 41.6$ ,  $41.6 < sec3 \leq 62.5$  and  $sec4 > 62.5$ .

Results showed how the adopted strategy was effective to control the swarm in its spatial distribution even in the case when the swarm is split into multiple groups.

Once a formation has been reached also the merging capability was tested and along with the assumption that each agent remembered its own local initial position the swarm was able to eventually reach its initial configuration.

## 4.1 Discussion of Simulations

In this Section the simulation results have been critically examined in the light of the previous state of the subject and judgements were made as to what has been learnt from this work. The proposed algorithm as a whole provides an effective way to interact with a multi-robot system. Moreover, it indicates a way towards a possible solution to overcome the major problems related to human-swarm interactions such as how this interaction should be structured, how to communicate with the whole group of robots and how to abstract the swarm out in such a way that could make sense for an user.

The few previous approaches to swarm control share the same *modus operandi* when trying to solve the problems deriving from that. Firstly, a notion of interaction dynamics is stated and this constrains the model of the system to be already partially defined in such a way that then harsher time is passed to figure out how this interaction dynamics can be effectively manipulated. The approach taken in this work followed the opposite direction with respect to all previous attempts by focusing on what constituted effective affordances for interacting with large collections of mobile robots. The goal of the entire work was to demonstrate the effectiveness of this new approach, so a *plethora* of improvements are left to do. As with any computer simulation, even this one when interacting with a multi-robot system, had certain weaknesses. Different considerations have to be taken with respect to the image recognition framework and the distributed swarm control framework.

Simulations showed how the image recognition framework turned out to be highly effective in the classification of the shapes present in the test images. As detailed in Section 3.1.1, an off-line computation of the SL is required. Both the alignment and the features extraction from the training shapes are phases that needed some time to be executed. A difficult choice

was to set the right parameters when extracting the features within a shape class, specifically the number of modes to consider after the eigenvalue decomposition of the shape variability matrix was performed. The image recognition algorithm, in the end, needed only few seconds to be executed and to provide the right classification.

On the side of the distributed swarm control framework, simulation results brought one key lesson derived from the use of local control laws. Since all agents defined the consensus values locally within their neighbors sets, this produced local variations in the reference orientation. The effects of these types of control laws were most visible in *stretched-like* shapes, and the most pronounced examples of those characteristics are depict in Figs. 4-1a, 4-1b, 4-5b, 4-5c, where the formations show non regular and wavy effects.

Slow convergence problems arise in the simulations since agents' inputs are driven by gradient descent based on their neighbors sets. By consequence, as the spatial distribution of the agents was expanding, also greater distances were reached between the agents reducing the cardinality of their neighbors sets. Noting that all agents' movements were asynchronous and independent from each other, a smaller neighbors set meant also smaller input and slower velocity. A solution for that could be to let the system to be time-varying so that the gains could change in such a way that they make sure the convergence velocity is improved. In accordance to that, the system dynamics would change and new proofs of convergence would be necessary.

In general, each framework lacked of a fast way to enlarge both the library of control laws and the shapes one. A considerable amelioration could be achieved by letting the libraries to define new shapes/control laws on-line by combining a set of the already defined ones and resulting in new possibilities for the entire swarm. Once a formation was reached, the merging control law had to be executed before that a new formation could have been demanded. This derives directly from the definition of the potential functions of Section 3.2 where agents changed their consensus values in part depending on what section they belonged to. For sake of clarity, the sections are those depict in Fig. 3-6.

Whenever two agents would have been spatially too close to each other, such potential functions did not make them to repel themselves, so the collision avoidance capability has been implemented to ensure collision-free movements. As result, the potential functions

generated control laws determined by two terms: an attraction consensus based term and a repulsion term. By consequence, another challenging phase was to choose the attraction and repulsion gains to promote smooth behaviors. Notice that since the agents' had a first order integrator dynamics, the problem of choosing the right combination of gains was not as hard as when dealing with more complex dynamics, inertia or other general constraints.





# Chapter 5

## Experimental Results

*“In theory, there is no difference between theory and practice. But in practice, there is.”*

**Yogi Berra**

Besides the theoretical models and the simulations tools adopted, a further step that has been taken was the experimentation with real robots. After all, various limits constrained the experimental phase. Firstly, the physical implementation is expensive and time consuming, but can bridge the gap between the theoretical problems one tries to solve and the real-world solutions for them. Other constraints were posed by the number of robots that constituted the swarm; it is generally impractical to have a swarm of robots for research purposes larger than few tens. A number of reasons motivate the experimental tests, maybe the main one is the current lack of experimental examples in swarm intelligence literature. In such an immature field of study, it is reasonable and more useful to rely mainly on strong foundation in theory without attempting any weak implementation. However, at this point, the field has matured so the further advances require both theory and practice combined. Another reason lays on the fact that the idea developed in this thesis to interact with multi-robot systems propose a brand new approach never taken before in literature. Previous attempts to human-swarm control started with the definition of the interaction dynamics and only then approaching the human-swarm interaction problem as one concerning how this interaction dynamics can be effectively manipulated.

A number of other methods have been proposed, with varying success. What it is presented

in this work takes a completely different perspective to solve the problem of controlling a swarm of robots by including the concept of affordances. This allowed the task to be far more intuitive and not affected by the number of agents in the swarm.

Simplicity is one of the pillars of swarm intelligence. Despite the simplicity of each individual, systems exhibited complex behaviors and these claims of simplicity seemed to suffice the exhibition of swarm intelligence. A secondary objective of the experimental phase is motivated by the desire to test the limitations of this principle of simplicity. The primary goals of experimentation follow from the motivations previously cited. The choices made regarding the experimental design were determined by the following goals:

- Develop a system capable of interacting with a multi-robot system and following in completion of the task of self-organizing in different formations.
- Introduce a novel approach to multi-robot systems interaction that is intuitive, not affected by the swarm size, does not require particular skills by the operator and has to be reliable and low-cost.



Figure 5-1: Khepera III.

## 5.1 Khepera III

In 1996 at the LAMI laboratory at EPFL (Lausanne, Switzerland) the first version of a differential wheeled mobile robot also known as Khepera was developed. Small, fast and architected around a Motorola 68331, it has been widely used by lots of researchers and universities worldwide. Since then, many extensions and new versions have been available. In this work a group of Khepera III robots, Fig. 5-1, has been used to verify, on real robots, the simulation results previously obtained. Further details on the Khepera III robot can be found in [29]. The way these mobile robots have been controlled is by the implementation of a software-based PI controller.

## 5.2 Deformable-Medium

The fascinating appeal taken in this work derives directly from the solution adopted to control the swarm, namely the control interface. The objective of the proposed algorithm



Figure 5-2: Clay interface representing two shapes examples: an *U-like* shape and a *Ball-shape*.

was to provide the operator with an interface for interacting with a multi-robot system and control the latter to assume a particular formation by molding clay, Fig. 5-2. This

solution allowed the interface to be low-cost, and reliable since it was not instrumented at all. Considering the case of heterogeneous swarms, from the interface point of view, this situation could be handled easily with clay of different colors.

### 5.3 Experimental Set Up

The experimental environment is described in this section. Each experiment started with an arbitrary initial spatial distribution of the group of robots. In all tests there was no maximum speed set for the Kheperas III, the only limit was the hardware/physical one. All experiments were conducted using a group of Khepera III robots introduced in Section 5.1. A camera was computer-controlled, and positioned above a high contrast surface where the clay had to be placed as illustrated in Fig. 5-3.

The only assumption needed is that the operator had to know in advance what shapes, and by consequence, what spatial distributions were achievable by the swarm. The previous assumption simply means that the operator had to be aware of the precomputed shape library SL.

Assuming that the number of control laws was always equal to the number of shapes for the matching problem to be determined, the developed framework allowed one to add as many shapes/control laws as desired with a two-steps process. This process required to update the SL and CLL with a new element respectively as detailed in Section 3.1.2. The communication between the computer and the agents is provided through a dedicated Wireless network. The agents' positions are supplied by an appropriate optical motion capture and tracking system based on a set of nine high speed motion capture cameras properly oriented to observe the scene where all the mobile agents were placed. The high accuracy, precision and updating rate of the tracking system allowed to have all the information needed during each step of the algorithm.

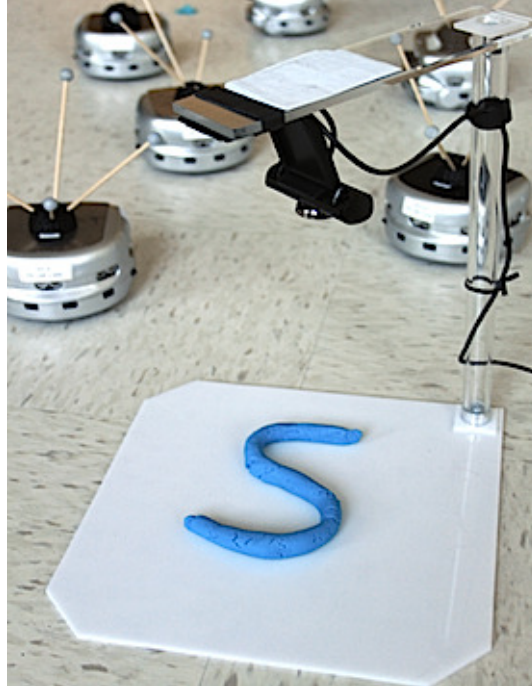


Figure 5-3: Experimental set up.

## 5.4 Experimental Cases and Results

Each experiment required to be initialized in such a way that the multi-robot system allowed the underlying communication graph  $G_{comm}$ , to be connected. By consequence, the agents' initial positions had to satisfy the communication constraints. Whenever a new shape has been detected the image recognition algorithm computed the appropriate match from the SL to the CLL, Fig. 3-1. Once the PC has computed the appropriate control law, the signal has been sent through Wireless network to a random agent in the swarm. Based on the assumption that  $G_{comm}$  was remaining connected all times, the signal was spread out to the whole swarm.

Figure 5-4 shows how the swarm successfully reproduced the desired *U-like* shape for the case of six robots. The parameters in this case were:  $\Delta = 0.9$ ,  $\delta = 0.13$ , *sensing range*  $R = 1\text{ m}$  and the bending sections were defined in meters and valued:  $0 < sec0 \leq 0.6$  and  $0.6 < sec1 \leq 0.8$  and  $sec2 > 0.8$ . The trajectories of the robots have also been considered and are shown in Fig. 5-5. The crosses in the figure represent the agents initial positions. The agents positions have been obtained by the optical motion capture and tracking system.

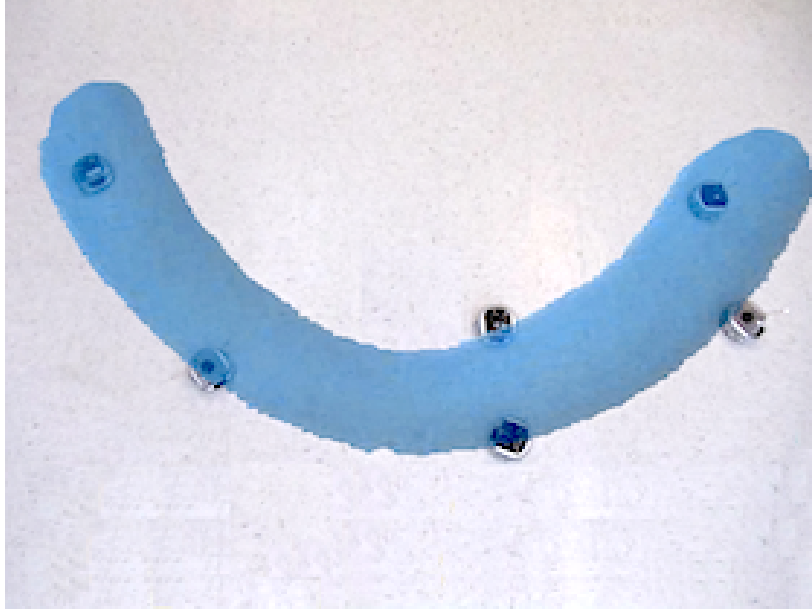


Figure 5-4: *U-like* shape superimposed on the swarm:  $\theta_{ref} = 0$ .

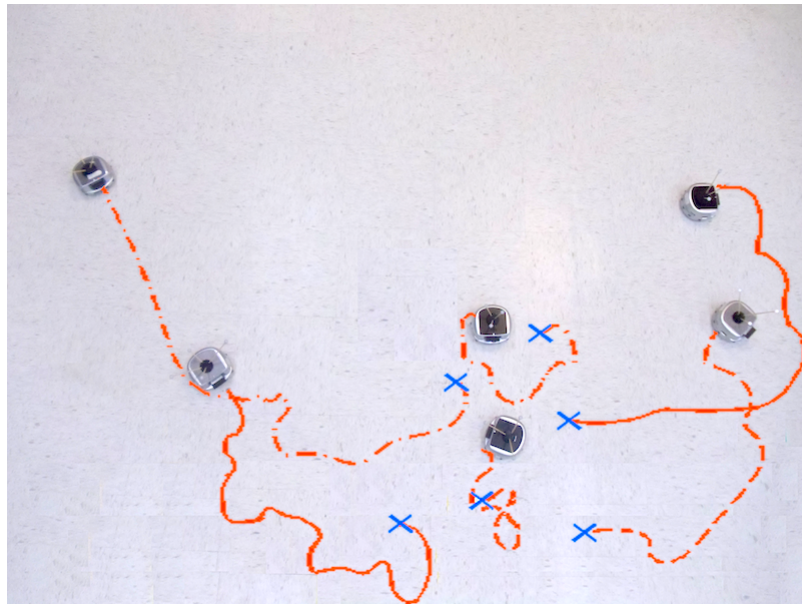


Figure 5-5: Trajectories of the swarm of Fig 5-4.

It is important to note that the purpose was not try to match the exact shape in the SL, but for example, perform a generic *U-like* shape instead. To clarify this consider that whenever an operator performed a shape (of the SL), the robots started to move after that the appropriate control law had been identified. The final positions, though, were dependent on the initial positions since each control law acted locally. As result, whenever an operator

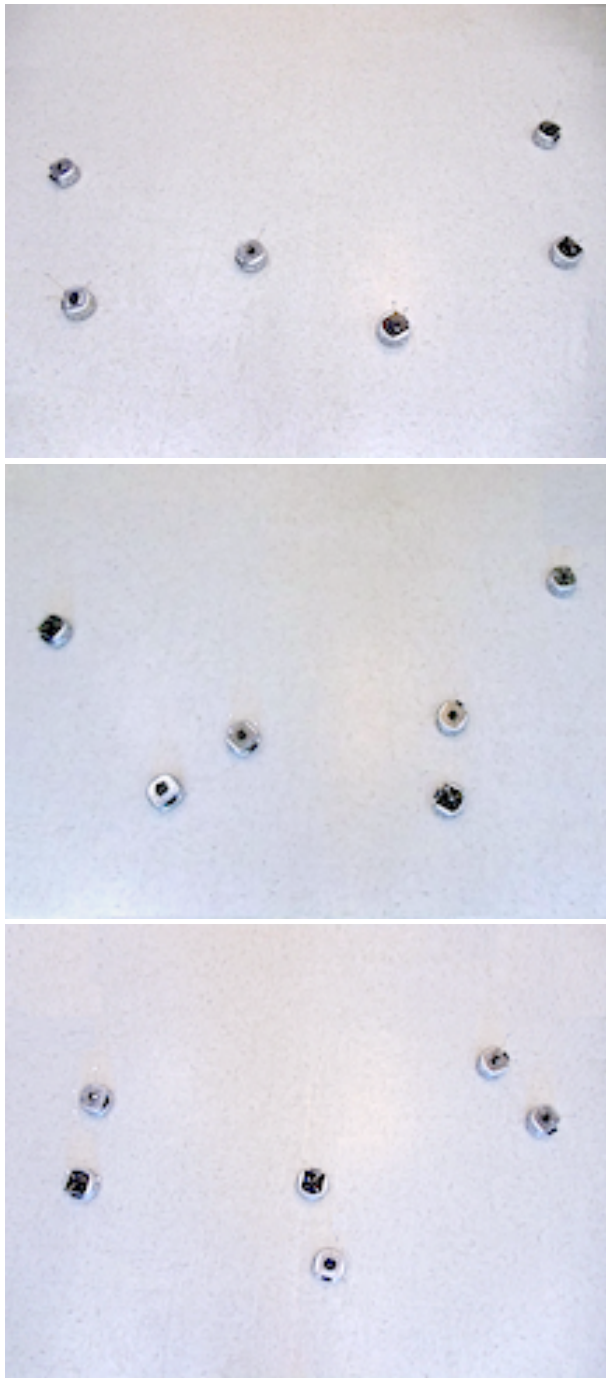


Figure 5-6: *U-like shapes*

wanted the swarm to perform a shape, i.e. *U-like* shape, the final spatial distribution could have been different each time. In Fig. 5-6 are depicted some *U-like* shapes obtained and although all formations are different from one other, the swarm successfully converged to an *U-like* shape each time.

The implementation of the algorithm on actual robots revealed the effectiveness of the idea in the restricted case of a small group of robots. The natural evolution of the followed approach expected that the next step was to scale up the system and verify again the algorithm behavior with new experiments. In accordance to that, the group of robots has been extended to a number of eleven robots. Following with the example of the *U-like* shape, Fig. 5-7 shows how the extended swarm successfully managed to perform the desired shape. The parameters in this case were:  $\Delta = 0.9$ ,  $\delta = 0.2$ , sensing range  $R = 1\text{ m}$  and the bending sections were defined in meters and valued:  $0 < sec0 \leq 0.6$ ,  $0.6 < sec1 \leq 0.75$ ,  $0.75 < sec2 \leq 0.95$  and  $sec3 > 0.95$ .

Even if the “quality” of the agents’ trajectories is not a parameter included in the goals of this implementation, the swarm trajectories are taken into account. The need of a transformation between the first order integrator dynamics to the differential wheel-drive dynamics of the actual robots and the inertia of the moving Kheperas III brought some serious challenges when dealing with the collision avoidance. In fact, the trajectories of the mobile robots are affected by non-optimal behaviors when the repulsion term ruled the interaction within an agent’s neighboring set, Fig. 5-8.

The same features previously exposed in the case of six robots emerged in the experiments on the extended swarm. Considering only one shape class, this means that depending on the robots’ initial positions, the formations the swarm converged to were always different to one other, nevertheless, each time the swarm managed to achieve the desired spatial distribution. A notable remark is again that the purpose of the distributed swarm control framework was to make the swarm to assume a generic shape belonging to the appropriate shape class and not exactly matching the one presented by the operator. Some results of *U-like* shapes obtained in different experiments are depict in Fig. 5-9.

Another parameter for evaluation of the proposed algorithm is the convergence time. The proposed method has been tested with different number of robots and the larger is the



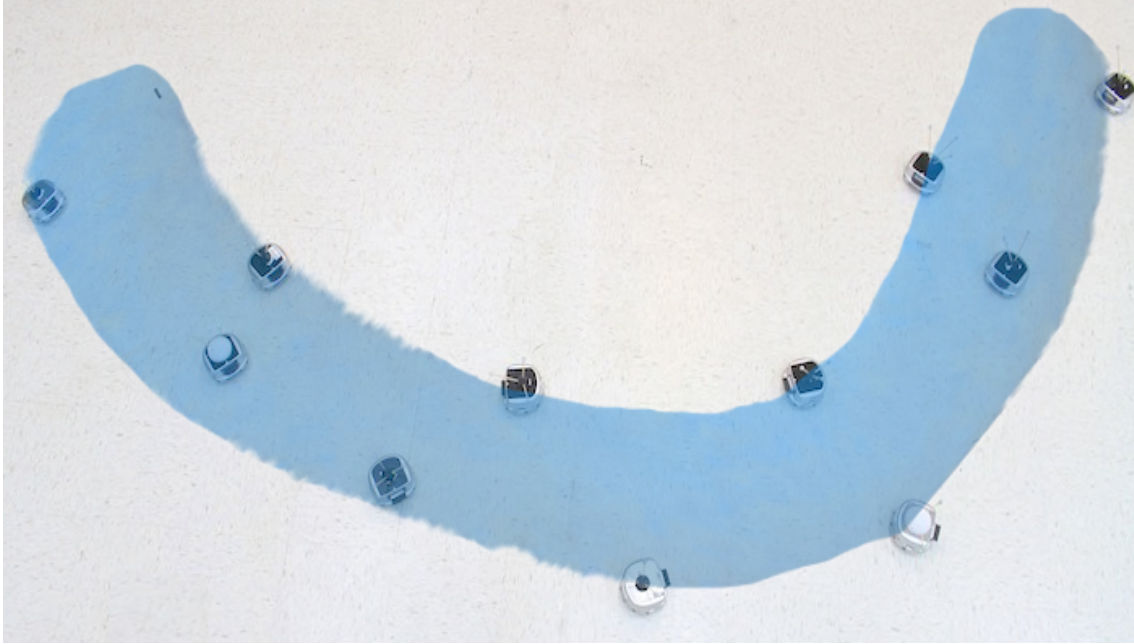


Figure 5-7: *U-like* shape superimposed on the extended swarm:  $\theta_{ref} = 0$ .

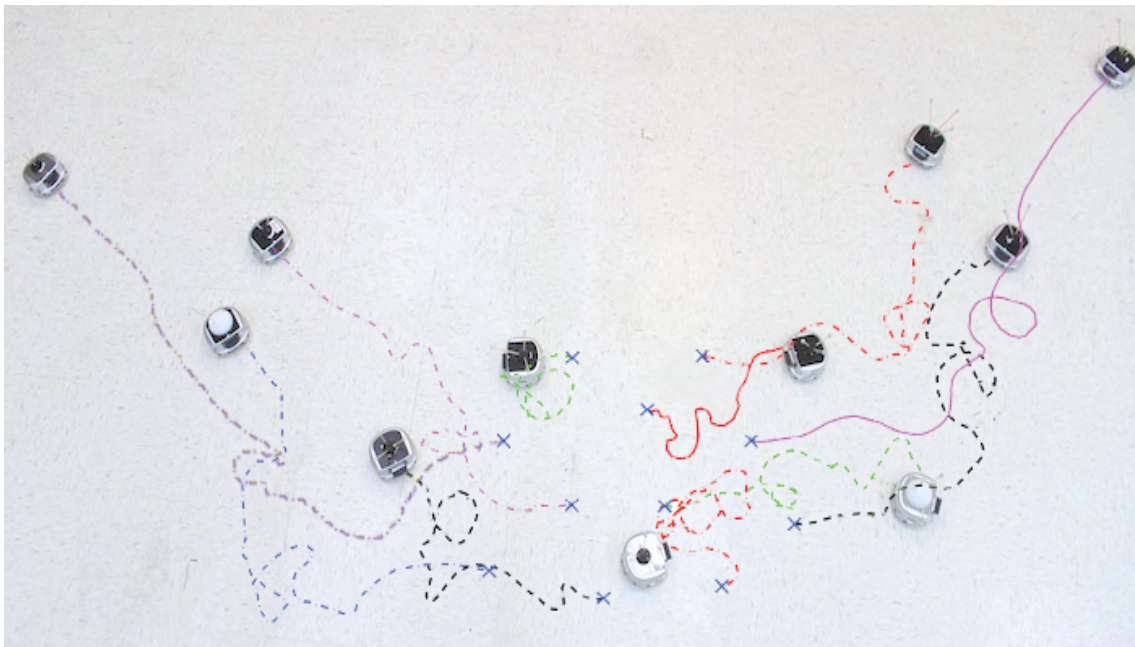


Figure 5-8: Trajectories of the extended swarm of Fig 5-7.

group, the greater the convergence time. In the case of six robots, the convergence times vary from 20 to 35 seconds depending on the shape that has to be performed meanwhile in the case of eleven robots the convergence times vary from 60 to 90 seconds. The greater bounds stand for the more complex shapes, e.g. splitting and/or bending cases.

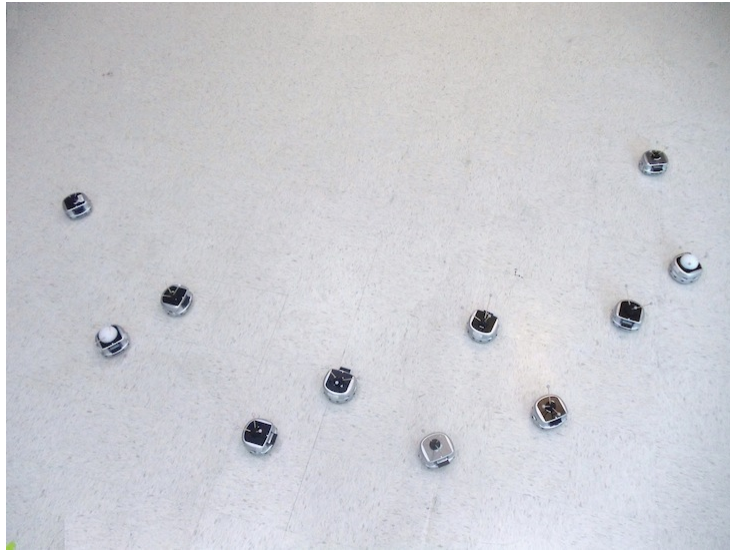
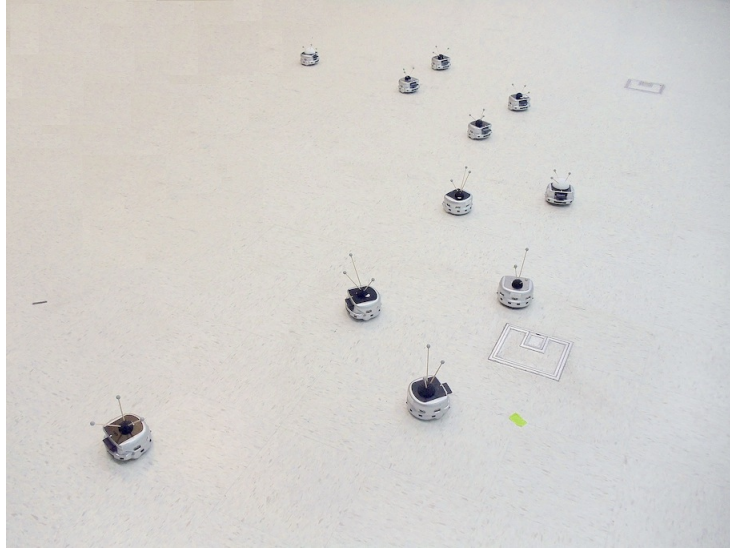


Figure 5-9: *U-like shapes*: extended swarm case.

## 5.5 Discussion of Experimental Results

In this Section the experimental results have been analyzed and considerations were made as to what has been obtained in the experiments. The general characteristics of the algorithm emerged during the simulations still held for the experimentation phase. The experimentation of the developed algorithm successfully allowed untrained operators to interact with the multi-robot system utilized by means of the proposed affordances. The strength of the encouraging results obtained lays in the validity of the followed idea which might pose a pillar towards the comprehension of how the interaction with multi-robot systems could be structured.

Further theoretical work is needed in this direction and most of the problems still remain unsolved; a critical one is how to accommodate the possibility to move the swarm once the desired formation is reached or how to comprehend some tasks capabilities beyond the ability of assuming those tight formations. Again, the goal of this work is to show a way where to address greater research efforts since this work takes clearly the form of an invitation rather than a solution for the human-swarm interaction problem.

A number of weaknesses were present in the operation of the experiments. Despite the general success of the experiments, there will always be possible improvements in a system of real robots. Perfection cannot be obtained while working in the real world and dealing with various sources of uncertainties. Due to spatial and other limitations, the scale of the experiments was limited to at most eleven robots. This number was adequate for demonstrating the elements in the set of the control laws. However, it is insufficient for thorough investigation of behaviors requiring large group size. Due to time constraints not all control laws were tested.

Other limitations have arisen from the model uncertainties of the robots and the transformation between the first order integrator dynamics, on which the control laws were defined, to the differential wheel-drive dynamics of the Kheperas III. The latter transformation required some fine tuning to allow a single agent to assume smooth-like behaviors when trying to follow inputs originally computed for a first order integrator model. It has not been an easy task to achieve such property.

All parameters had to change with respect to the simulation phase, moreover some of them acquired physical meaning due to the experimental elements involved in the set up. Due to the sensitivity of the implementation, the collision avoidance term in the potential functions of Section 3.2 has been changed to a different repulsion function. As the scale of the group of agents increased, the attraction and repulsion gains had to be significantly decreased to preserve the stability of the system.

The experimental results, in the end, provided enough proofs in favor of the proposed algorithm.

# Chapter 6

## Conclusions

*“People love chopping wood. In this activity one immediately sees results.”*

**Albert Einstein**

The objective of this thesis was to test the effectiveness of a novel approach for interacting with multi-robot systems. Untrained operators should have been able to control large collection of mobile agents, so a decentralized and distributed approach was taken to solve the problem.

The interaction method selected to stimulate the group of robots is by means of the definition of affordance. The affordances we identify when controlling a swarm, include stretching the swarm, molding it into a particular shape, splitting and merging sub-swarms, and mixing of different swarms. One object that provides such affordances is clay, which is a physical object that it is easy to manipulate. As such, the main contribution of this thesis is the identification of the clay as an effective multi-agent control interface, and the accompanying algorithms needed to identify the clay shapes, and then turn those into distributed control laws for a team of mobile robots.

From the architecture point of view, two main frameworks have to be distinguished – an image recognition framework and a distributed swarm control framework. The image recognition framework had to provide the right match between two off-line created libraries, namely the Shapes Library and the Control Laws Library. Results showed how the developed recognition phase was highly successful, fast and robust to provide such clas-

sification even in cases where the operator performed non regular and non uniform shapes. The distributed swarm control framework was developed without a global supervisor, but on the contrary, allowed only limited local information to be available. The single mobile agents could reach the final objective by relying only on the collaboration and coordination within the swarm. In this way complex behaviors and improved capabilities can emerge from such simple entities.

Different cases were designed and both simulated and tested to verify the validity of the idea. In such novel approach, there are no limitations in the possible extensions and improvements left to be done. For sake of completeness, some of the main ones are:

- develop more complex shapes, which is the most trivial one.
- Include the capability of moving the swarm once a formation is reached.
- Allow some task capability beyond the spatial self-organization.
- Improve more scalable communication between the operator and the swarm once the appropriate control law has been identified.
- Speed up the convergence rate of the swarm.
- Reach deep theoretical advances before facing any real implementation.

As result, this work provides a clear proof of the effectiveness of the proposed method for interacting with multi-robot systems. Nevertheless, further theoretical advances are required to found the presented idea as to be a solution for interacting with multi-robot systems.

# Bibliography

- [1] J. Fax, and R. Murra. Information flow and cooperative control of vehicle formations. IEEE Transactions on Automatic Control, 49(9):1465-1476, 2004.
- [2] M. Mesbahi and M. Egerstedt, Graph Theoretic Methods in Multiagent Networks, Princeton University Press, NJ, 2010.
- [3] J.P. de la Croix and M. Egerstedt. Controllability Characterizations of Leader-Based Swarm Interactions. AAI Symposium on Human Control of Bio-Inspired Swarms, Arlington, DC, Nov. 2012.
- [4] P. Kingston and M. Egerstedt. Distributed-Infrastructure Multi-Robot Routing using a Helmholtz-Hodge Decomposition. IEEE Conference on Decision and Control, Orlando, FL, Dec. 2011.
- [5] N. Ghods and M. Krstic. Multiagent deployment over a source, IEEE Transactions on Control Systems Technology, vol. 20, pp. 277-285, 2012.
- [6] R. Arkin and K. Ali. Integration of reactive and telerobotic control in multi-agent robotic systems. Proc. Third International Conference on Simulation of Adaptive Behavior, pages 473-478, 1994.
- [7] A. Atherton and M. Goodrich. Supporting Remote and Mobile Manipulation with an Ecological Augmented Virtuality Interface, Proceedings of AISB-HRI Symposium New Frontiers in Human-Robot Interaction, Scotland, 2009.

- [8] B. Hardin and M. A. Goodrich. On Using Mixed-Initiative Control: A Perspective for Managing Large-Scale Robotic Teams, Proceedings of ACM/IEEE International Conference on Human-Robot Interaction, San Diego, CA, March 2009.
- [9] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, and B. Schmidt. Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. AAAI Spring Symposium, 2006.
- [10] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, and B. Marinier. Progress towards multi-robot reconnaissance and the MAGIC 2010 Competition, Journal of Field Robotics, Vol. 29, No. 5, pp. 762792, 2012.
- [11] J.J. Gibson. *The Theory of Affordances*. In *Perceiving, Acting, and Knowing*, Eds. Robert Shaw and John Bransford, 1977.
- [12] Uexkll, Jakob von (1980 [1920 etc.]): *Kompositionslehre der Natur*. (Edited by Thure von Uexkll). Frankfurt am Main.
- [13] Donald Norman, *The Design of Everyday Things*, ISBN 0-465-06710-7. Originally published under the title *The Psychology of Everyday Things*, often abbreviated to POET.
- [14] Parrish, J. K., Viscido, S., and Grunbaum, D. 2002. Self-organized fish schools: An examination of emergent properties. *Biol. Bull.* 202:296 - 305.
- [15] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model, in *Computer Graphics*, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34, 1987.
- [16] Fukada, T., Nakagawa, S., Kawauchi, Y., and Buss, M., "Structure decision method for self-organizing robots based on cell structure—CEBOT," presented at IEEE International Conference on Robotics and Automation, Los Alamitos, CA, pp. 695-700, 1989.



- [17] Beni, G., "The concept of cellular robot," presented at Third IEEE Symposium on Intelligent Control, Arlington, VA, pp. 57-61, 1988.
- [18] Brooks, R. A., "A robust layered control system for a mobile robot," *Journal of Robotics and Automation*, vol. 2, pp. 14-23, 1986.
- [19] E. Sahin, M. Dorigo. Special issue:swarm robotics. *Autonomous robots*, 2004.
- [20] Erol Sahin, Sertan Girgin, Levent Bayindir, Ali Emre Turgut. *Swarm robotics*. 2008. *Swarm Intelligence: Introduction and Applications. Natural Computing Series* pages 87-100
- [21] A. Tsai, A. J. Yezzi, and A. S. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Trans. on Medical Imaging*, 22(2):137 - 154, 2003.
- [22] S. Osher and J. Sethian, Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, vol. 79, pp. 12 - 49, 1988.
- [23] G. Borgefors, "Distance transformations in digital images," *CVGIP:ImageUnderstanding*, vol.34, pp. 334 - 371, 1986.
- [24] M. Leveton, Statistical models in medical image analysis, Ph.D. dissertation, Massachusetts Inst. Technol, Dept. Elect. Eng., 2000.
- [25] T. Chan and L. Vese, Active contours without edges, *IEEE Trans. Image Processing*, vol. 10, pp. 266 - 277, Feb. 2001.
- [26] N. Paragios and R. Deriche, Geodesic Active Regions for Texture Segmentation, INRIA, Sophia Antipolis, France, Res. Rep. 3440, 1998.
- [27] R. Ronfard, Region-based strategies for active contour models, *Int. J. Comput. Vis.*, vol. 13, pp. 229 - 251, 1994.
- [28] A. Yezzi, A. Tsai, and A. Willsky, A statistical approach to snakes for bimodal and trimodal imagery, in *Proc. Int. Conf. Computer Vision*, vol. 2, 1999, pp. 898 - 903.

[29] Mondada, F., Franzi, E., Guignard, A. (1999), The Development of Khepera. In proceedings of First International Khepera Workshop, Paderborn, 10-11 December 1999.