



Revista Facultad de Ingeniería Universidad de Antioquia

ISSN: 0120-6230

revista.ingenieria@udea.edu.co

Universidad de Antioquia
Colombia

Moreno Arboleda, Francisco Javier; Duitama Muñoz, John Freddy; Ospina, Edison Camilo
A method for estimating the position and direction of a leader of a set of moving objects
Revista Facultad de Ingeniería Universidad de Antioquia, núm. 62, enero-marzo, 2012, pp. 11-20
Universidad de Antioquia
Medellín, Colombia

Available in: <http://www.redalyc.org/articulo.oa?id=43025115002>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

A method for estimating the position and direction of a leader of a set of moving objects

Un método para estimar la posición y la dirección del líder en un conjunto de objetos móviles

Francisco Javier Moreno Arboleda^{1*}, John Freddy Duitama Muñoz², Edison Camilo Ospina¹

¹Universidad Nacional de Colombia. Facultad de Minas. Carrera 80 No. 65-223 Bloque M8A. Medellín, Colombia.

²Universidad de Antioquia. Facultad de Ingeniería. Calle 67 N.º 53-108 Bloque 21 Lab. 316. Medellín, Colombia.

(Recibido el 25 de agosto de 2011. Aceptado el 27 de febrero de 2010)

Abstract

Movement patterns can be identified when studying a group of moving entities such as a group of people, a flock of birds, a school of fish, a convoy of vehicles, among others. In this paper, it is analyzed a pattern, known as *leadership*. Informally, this pattern is characterized by a moving entity called *leader* that motivates or represents the behavior of the group in order to reach a goal during a period. A formal method is proposed to estimate the position and the direction where a leader should be located and headed at a time-point in order to lead a group. These estimations can also be useful to check the consistency of the data about a leadership pattern, and to estimate the missing information (position and direction) of a leader at a specific time, i.e., an *imputation process*. In order to show the expediency of the proposal, a series of experiments were implemented and conducted using Netlogo, a programmable modeling environment for simulating natural and social phenomena.

----- **Keywords:** Moving objects, movement patterns, flocks, leadership

Resumen

Patrones de movimiento pueden ser identificados cuando se estudia un grupo de entidades móviles, como un grupo de personas, una bandada de pájaros, un banco de peces, un convoy de vehículos, entre otros. En este artículo,

* Autor de correspondencia: teléfono: + 57 + 4 + 425 5376, fax: + 57 + 4 + 425 5365, correo electrónico: fjmoreno@unal.edu.co (F. Moreno)

se analiza un patrón, conocido como *liderazgo*. Informalmente, este patrón se caracteriza por una entidad móvil llamada *líder* que motiva o representa el comportamiento de un grupo con el fin de alcanzar un objetivo durante un período. Se propone un método formal para estimar la posición y la dirección donde un líder debería estar ubicado y orientado en un punto del tiempo con el fin de liderar un grupo. Estas estimaciones pueden también ser útiles para verificar la consistencia de los datos de un patrón de liderazgo y para estimar la información faltante (posición y dirección) de un líder en un tiempo específico, i.e., un *proceso de imputación*. Con el fin de mostrar la conveniencia de la propuesta, se implementó y se desarrolló una serie de experimentos mediante Netlogo, un entorno programable de modelado para la simulación de fenómenos naturales y sociales.

----- **Palabras clave:** Objetos móviles, patrones de movimiento, bandadas, liderazgo

Introduction

Movement patterns can be identified when studying a group of moving entities such as a flock of birds, a school of fish, a convoy of vehicles, among others [1, 2]. A pattern, known as *leadership* has been recognized in [3]. This pattern is characterized by a moving entity called *leader* that motivates or represents the behavior of the group in order to reach a goal during a period. A leadership pattern can be defined using some temporal constraints and a geometrical arrangement between the leader entity and the other entities in the group, called *followers*.

Given a group of moving entities and a time-point t , a method is proposed to estimate the position and the direction where a leader should be located and headed at t in order to lead this group. This problem has been identified in a short oral communication [4]. The estimations are based on the leadership pattern definition proposed by Andersson [3].

The method may be applied to locate and destroy the leader of a troop in military operations [5] or in games such as Battlefield and Squad Leader, and to locate the lead robot for a dissociated trailing robot [6] or for an animal that fell behind in a flock.

Motivating problems

Problem 1: Finding a Position/Direction for a Leader

Consider a convoy of ships at a time t (figure 1). The position and direction of each ship is known at t . The idea is to determine the position and direction of a convoy commander at t to lead this group. Note that the position and direction of the commander must consider the current position and direction of each ship (follower) of the convoy, so that each follower can “see” (perceive) and follow the commander. Figure 1 shows a convoy of ships and the position of a possible leader (represented as a dashed ship).

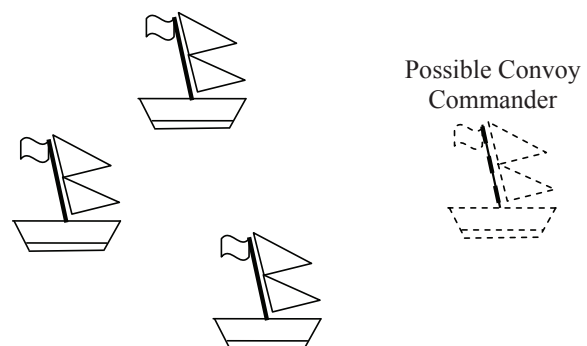


Figure 1 A convoy of ships that requires a commander (dashed ship) at time t

Problem 2: Consistency Checking

Suppose there are data related with the position and the direction of a leader entity and its followers during a leadership period (table 1). If the position and direction of the leader are constrained by the position and direction of its followers, the data can be checked in order to identify possible inconsistencies. For example,

assume that a leader should always be *in front of* its followers. Therefore, the data at time t_3 in table 1 would be wrong because these data indicate that the leader was *behind* of its followers. Once inconsistencies are identified, the goal is to try to correct them. Note that even in a small data sample (five records), the detection of this type of inconsistencies is not evident.

Table 1 Sample data of a leader and its followers during an interval $[t_1, t_5]$

<i>Time-point</i>	<i>Leader Position (x, y)</i>	<i>Followers Position</i>
t_1	(10,20)	{(7,20),(8,19),(8,21),(9,19),(9,21)}
t_2	(15,20)	{(12,20),(13,19),(13,21),(14,19),(14,21)}
t_3	(16,20) → <i>Inconsistent data</i>	{(17,20),(18,19),(18,21),(19,19),(19,21)}
t_4	(25,20)	{(22,20),(23,19),(23,21),(24,19),(24,21)}
t_5	(30,20)	{(27,20),(28,19),(28,21),(29,19),(29,21)}

Problem 3: Leader's Data Imputation

Consider again table 1 and suppose that the data for the leader position at time t_3 is missing (table 2). Suppose that there was a leadership pattern during the interval $[t_1, t_5]$. In addition, the

constraints of position and direction of the leader with regard to the position and direction of each of its followers are also known. The idea is to try to estimate missing data of the leader, i.e., to impute its values [7].

Table 2 Missing data of a leader

<i>Time-point</i>	<i>Leader Position (x, y)</i>	<i>Followers Position</i>
t_1	(10,20)	{(7,20),(8,19),(8,21),(9,19),(9,21)}
t_2	(15,20)	{(12,20),(13,19),(13,21),(14,19),(14,21)}
t_3	<i>Missing data</i>	{(17,20),(18,19),(18,21),(19,19),(19,21)}
t_4	(25,20)	{(22,20),(23,19),(23,21),(24,19),(24,21)}
t_5	(30,20)	{(27,20),(28,19),(28,21),(29,19),(29,21)}

The original model

Next, the essential elements of Andersson's model [3] are presented. Consider a set E of n entities $\{e_1, e_2, \dots, e_n\}$ moving in a space, usually

a geographic region, during an interval $[t_i, t_f]$. This space is represented by the Euclidean plane. The time is represented in a continuous form. Tp denotes the infinite set of time-points: $\{t \mid t \in [t_i, t_f]\}$. On the other hand, Ts denotes the

discrete set of time-points $\{t_1, t_2, \dots, t_j\}$. Each $t_i \in Ts$, represents a *time-step* and corresponds to the time-point in which the position (and possibly other data) of the moving entity was recorded. (t_{i-1}, t_i) , $t_i \in Ts$, $i \neq 1$; represents a *unit-time-interval*. The size of (t_{i-1}, t_i) is equal to the size of (t_{j-1}, t_j) , $\forall t_j \in Ts, j \neq 1$.

The coordinates of an entity at a time-point t are given by a pair of functions $xpos$ and $ypos$, both with signature: $E \times Tp \rightarrow \text{Real}$. It is assumed that between two consecutive time-steps, the entity moves along a straight line and with constant velocity [3]. Let t be a time-point and $t_x \in Ts$, $x \neq 1$; the angle of e_i , denoted $d(e_i)$, at t is defined by the line segment that goes from $(xpos(e_i, t_{x-1}), ypos(e_i, t_{x-1}))$ to $(xpos(e_i, t_x), ypos(e_i, t_x))$, where $t_{x-1} < t < t_x$ (figure 2). The angle $d(e_i)$ is between $[0, 2\pi)$.

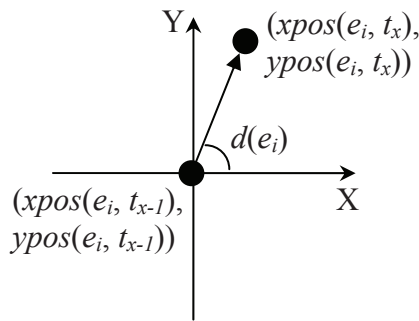


Figure 2 Angle of e_i : $d(e_i)$ during $[t_{x-1}, t_x]$

The *front-region* of an entity e_i is a region associated with e_i at a time-point t ($t \in Tp, t \notin Ts$) that represents the region of perception of the entity, e.g., its visual range. The front-region of e_i is defined as follows: consider three line segments s_0, s_1 , and s_2 , each one of length r . Each segment has an endpoint in the position $(xpos(e_i, t), ypos(e_i, t))$. The direction of the segment s_0 is equal to the angle $d(e_i)$ at t . The segments s_1 and s_2 form angles of $\alpha/2$ and $-\alpha/2$ ($\alpha \leq 2\pi$) with regard to the segment s_0 , respectively. The circular sector (wedge-shaped region) of radius r , bounded by s_1 and s_2 , forms the *front-region* of e_i at t and it is denoted $front(e_i)$ (figure 3).

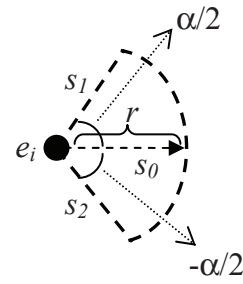


Figure 3 Front-region of e_i

An entity e_j is *in front of* an entity e_i at a time-point t ($t \in Tp, t \notin Ts$), if e_j is in the front-region of e_i , i.e., $(xpos(e_j, t), ypos(e_j, t))$ is inside $front(e_i)$; this is written $e_j \in front(e_i)$ and it is said that e_i *follows* e_j (e_i is a *follower* of e_j) at t . An additional requirement to state that e_i follows e_j can be enforced: let $\beta \in [0, \pi]$, the entity e_i follows e_j at t if i) $e_j \in front(e_i)$ at t and ii) $\|d(e_i) - d(e_j)\| \leq \beta$, where $d(e_i)$ and $d(e_j)$ are the directions of e_i and e_j at t (figure 4). An entity e_i *follows* an entity e_j during an interval $[t_a, t_b]$, t_a and t_b time-points, if and only if e_i follows e_j at t , $\forall t \in [t_a, t_b]$, $t \notin Ts$. Note that, if e_i follows e_j at t , where t is a time-point between two consecutive time-steps, i.e., $t_{x-1} < t < t_x$, ($t_x \in Ts, x \neq 1$) then it holds that e_i follows e_j during any time-point in (t_{x-1}, t_x) and; therefore, during any subinterval of (t_{x-1}, t_x) .

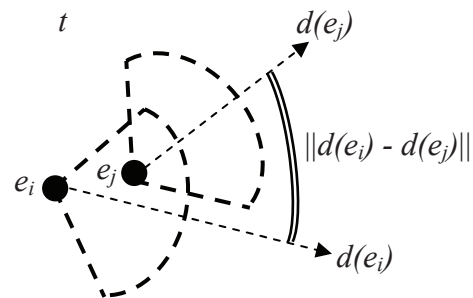


Figure 4 $e_j \in front(e_i)$. If $\|d(e_i) - d(e_j)\| \leq \beta$ then e_i follows e_j

Finally, the concepts of *leader* and *leadership pattern* are presented. An entity e_i is a *leader* during an interval $I = [t_a, t_b]$ if: i) e_i does not follow any entity during I and ii) e_i is followed by a number m of entities at each time-point of I . There

is a *leadership pattern* if e_i has at least m followers at each time-point of I and the size of I is at least

k unit-time-intervals. Figure 5 shows an entity e leader of three entities during an interval $[t_a, t_b]$.

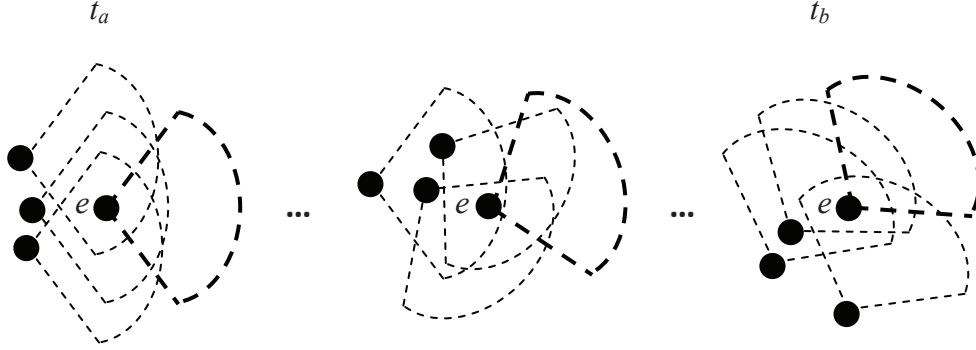


Figure 5 An entity e leader of three entities during $[t_a, t_b]$

Estimating the position and direction of a leader

Estimation of the leader position

Let $(xpos(e_{leader}, t), ypos(e_{leader}, t))$ be the position of a leader entity e_{leader} at time t with regard to a group of followers F . Because each entity $e_i \in F$ must be a follower of e_{leader} then $e_{leader} \in front(e_i) \forall e_i \in F$. That is, the position of e_{leader} at t must be contained in the front region of each $e_i \in F$: $(xpos(e_{leader}, t), ypos(e_{leader}, t)) \in front(e_i), \forall e_i \in F$. Therefore, if e_{leader} is a leader at t for the entities in F then e_{leader} must be located in a region Ω , i.e., the intersection of the front regions of each entity in F : $(xpos(e_{leader}, t), ypos(e_{leader}, t)) \in \Omega$ (figure 6).

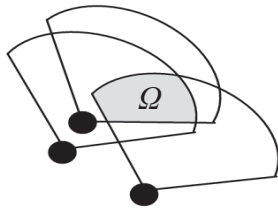


Figure 6 Intersection region Ω (in grey)

Using the associative property of the intersection operator (n-fold intersection) [8], the region Ω for n entities can be expressed as:

$$\Omega = \bigcap_{i=1}^{i=n} front(e_i), e_i \in F \quad (1)$$

Note that if $\Omega = \emptyset$ then it is not possible to find a position for a leader with regard to the set of entities in F . In figure 7, an algorithm is presented to find the intersection Ω of the front regions of n entities at time t . The order of the algorithm is $O(n)$. The center of mass of Ω can be an estimation for the position of e_{leader} .

Algorithm FrontRegionIntersection

Input: A set of n entities: $F = \{e_1, e_2, \dots, e_n\}$
A time-point t

Output: Intersection of the front regions at time t : Ω

```

1. BEGIN
2.  $\Omega \leftarrow front(e_1)$  // Front region of  $e_1$  at time  $t$ 
3. FOR  $i \leftarrow 2$  TO  $n$ 
4.  $\Omega \leftarrow \Omega \cap front(e_i)$  // Front region of  $e_i$  at time  $t$ 
5. IF  $\Omega = \emptyset$  THEN // If intersection becomes empty
6. BREAK // Exit loop
7. END IF
8. END FOR
9. RETURN  $\Omega$ 
10. END

```

Figure 7 Algorithm 1: Intersection of the front regions of n entities at time t

Example 1 Consider a set of three entities with their respective front regions at time t as shown in

figure 8a. In line 2, the Algorithm 1 takes $front(e_1)$ as the intersection region Ω as is shown in figure 8b (bold shape). Next, the following steps are performed: 1) It is found the intersection between

Ω and $front(e_2)$ at t (figure 8b). 2) It is found the intersection between Ω and $front(e_3)$ at t (figure 8c). 3) Loop finishes and Ω is returned (bold shape in figure 8d).

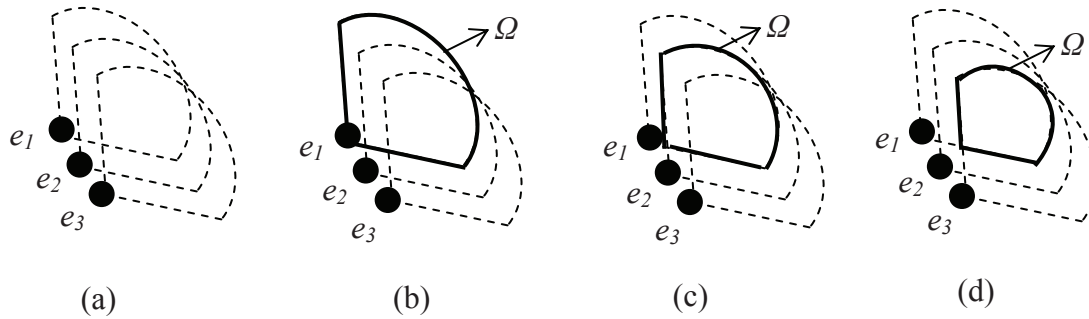


Figure 8 Example of Algorithm: non-empty intersection of front-regions of a set of three entities

Estimation of the direction of a leader

in addition to estimate the position where a leader must be located at time t with regard to a set of followers, the challenge is also to estimate its direction considering the constraints presented (figure 4).

Let $d(e_{leader})$ the direction of an entity e_{leader} at time t . If an entity $e_i \in F$ is a follower of e_{leader} then $\|d(e_i) - d(e_{leader})\| \leq \beta$. This condition can be written as $d(e_{leader}) \in [d(e_i) - \beta, d(e_i) + \beta]$. That is, if e_{leader} is a leader at time t for the set of entities in F then the direction of e_{leader} must be included in the interval defined by each entity in F . This is an analogous problem to the one where the position for e_{leader} was estimated. Therefore, $d(e_{leader})$ must be included in the intersection ∇ of the intervals defined by the entities in F (figure 9): the value of $d(e_{leader})$ must be included in the interval delimited between the dashed lines.

Let $Mind$ and $Maxd$ be the minimum and the maximum of the directions of the entities in F , respectively. ∇ is a non-empty interval if the intersection between the intervals defined by $Mind$ and $Maxd$ is not empty. That is, if $Maxd - \beta \leq Mind + \beta$. Therefore, the intersection of these intervals is non-empty if $Maxd \leq Mind + 2\beta$,

giving rise to an intersection interval ∇ : $[Maxd - \beta, Mind + \beta]$. Similarly to the problem of the intersection of front regions, if $Maxd > Mind + 2\beta$ then it is not possible to find a direction for a leader with regard to the set of entities in F . In figure 10 an algorithm is presented to find the intersection of the intervals of n entities at a time-point t . The order of this algorithm is also $O(n)$. An estimation for the direction of e_{leader} is the midpoint of the intersection interval ∇ : $d(e_{leader}) = (Maxd - \beta + Mind + \beta) / 2 = (Maxd + Mind) / 2$.

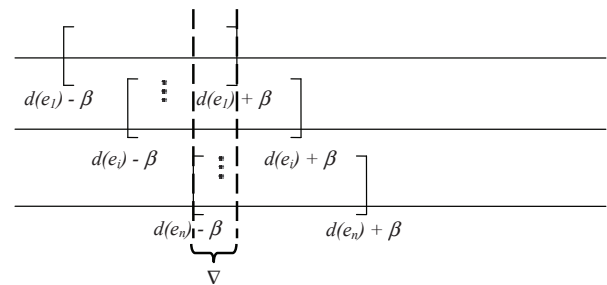


Figure 9 Intersection ∇ of n intervals

Example 3 Consider a set of three entities with their respective directions (figure 11a). Suppose that the leader must maintain a difference of direction with regard to its followers equal to $\beta = \pi/2$. Therefore, the intervals defined by the

directions of e_1 , e_2 , and e_3 are $d(e_1) = \pi/3$ then $[\pi/3 - \pi/2, \pi/3 + \pi/2] = [-\pi/6, 5\pi/6]$; $d(e_2) = \pi/4$ then $[\pi/4 - \pi/2, \pi/4 + \pi/2] = [-\pi/4, 3\pi/4]$; $d(e_3) = \pi/6$ then $[\pi/6 - \pi/2, \pi/6 + \pi/2] = [-\pi/3, 2\pi/3]$.

Because $Mind = \pi/6$ and $Maxd = \pi/3$ then $\pi/3 \leq \pi/6 + 2(\pi/2)$; therefore, the intersection interval ∇ is $[-\pi/6, 2\pi/3]$ (figure 11b).

DirectionIntersection

Input: A set of n entities: $F = \{e_1, e_2, \dots, e_n\}$

A time-point t

Difference of direction threshold: β

Output: Interval Intersection at time t : ∇

1. **BEGIN**
2. $Mind \leftarrow$ Minimum direction of the entities in F at time t
3. $Maxd \leftarrow$ Maximum direction of the entities in F at time t
4. **IF** $Maxd \leq Mind + 2*\beta$ **THEN**
5. $\nabla \leftarrow [Maxd - \beta, Mind + \beta]$
6. **ELSE**
8. $\nabla \leftarrow \emptyset$
9. **END IF**
10. **RETURN** ∇
11. **END**

Figure 10 Algorithm 2: Intersection of the intervals generated by the directions of a set of n entities at time t

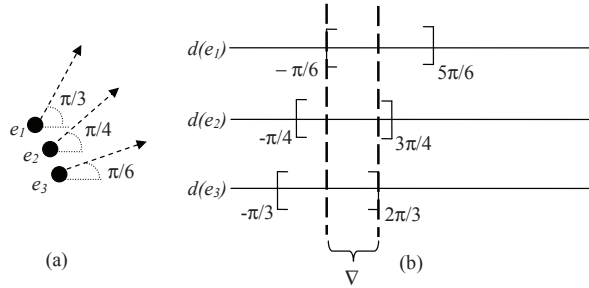


Figure 11 Example of Algorithm: non-empty intersection of three intervals defined by the directions of three entities

Experiments

To show the expediency of the proposal, the algorithms were implemented and a set of experiments on various data sets were conducted. The algorithms were implemented using Oracle Spatial features [9].

On the other hand, the obtention of real data for the experiments was difficult. Therefore, the data collection was simulated using the *flocking model* built in the Netlogo Library [10]; an approach that has been followed in previous works [3, 11, 12].

This flocking model is based on three rules: *alignment*, which establishes the tendency of an entity to turn in the same direction that its nearby entities; *separation*, which express that an entity will turn to avoid another entity which is getting too close, and *cohesion*, which means that an entity will move towards its nearby entities. The Netlogo simulation module also defines additional parameters to influence the behavior of the entities. Some of them are *vision* which represents the radius r of vision of an entity, *minimum-separation* which establishes the minimum distance between two entities in the space, and three turn-angles defined by *max-align-turn*, *max-cohere-turn*, and *max-separate-turn* to control the maximum angle that an entity can turn.

Appropriate modifications were incorporated to Netlogo in order to detect leadership patterns, e.g., parameters such as β , α , and *min-num-followers* (which represents the minimum number of followers to consider an entity as a leader of a group during a period) were included.

To evaluate the algorithms **FrontRegionIntersection** and **DirectionIntersection** the following approach was applied: for each leadership pattern that was found, the algorithms were applied to estimate the position and direction of the leader at a specific time-point; then, these results were compared with the simulated ones.

Table 3 shows an example: *min-num-followers* was set to 5 during $[t_1, t_3]$, and the data of a leader at t_3 were estimated. To compare the simulated results with the calculated ones, the following actions were considered: i) positions were compared using the distance between two points. For example, the distance between the simulated position (43, 91) and the estimated position (39.99, 91.93) was 3.1, and ii) directions were compared using the absolute value of the

difference between the simulated angle and the calculated one. For example, the difference

between 230.69 and 230.48 was 0.23 degrees. Figure 12 shows the intersection region at t_3 .

Table 3 Data of a leadership pattern generated by Netlogo and estimated data at time-point t_3

Time-point	Leader		Followers		
	Position (x, y)	Direction	Follower	Position (x, y)	Direction
t_1	(45,93)	231,78	e_1	(41,110)	244,61
			e_2	(51,99)	226,66
			e_3	(57,95)	220,39
			e_4	(45,109)	232,12
			e_5	(47,99)	231,72
t_2	(44,92)	230,98	e_1	(41,99)	244,61
			e_2	(50,98)	224,93
			e_3	(56,94)	220,39
			e_4	(44,108)	232,12
			e_5	(47,99)	231,72
t_3	(43,91) Estimated: (39.99, 91.93)	230,48 Estimated: 230,69	e_1	(40,109)	242,61
			e_2	(50,97)	223,17
			e_3	(55,93)	218,79
			e_4	(44,108)	229,58
			e_5	(46,98)	230,06

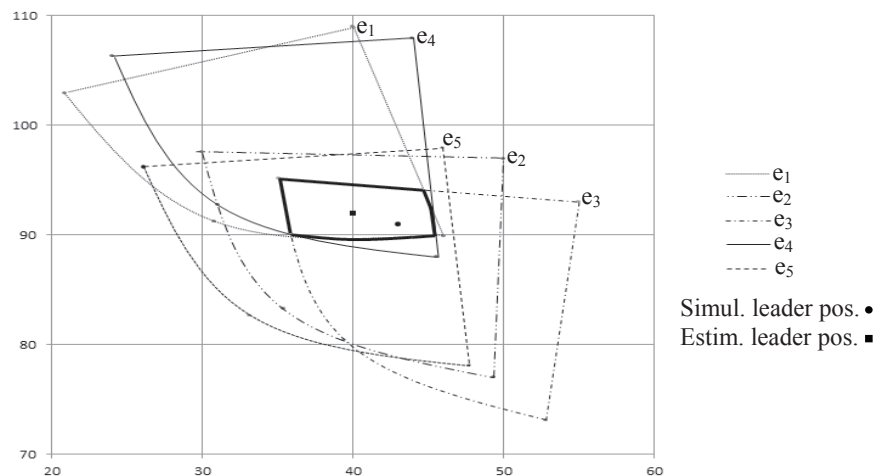


Figure 12 Intersection region at time-point t_3 (data generated by Netlogo)

The experiments were conducted with different numbers of followers ($0 < m < 11$), $\beta = \pi/2$, $r = 20$, $\alpha = \pi/2$, and *minimum-separation* = 3. The *max-align-*

turn, *max-cohere-turn*, and *max-separate-turn* angles were set to 5, 3 and 1.5 degrees, respectively. Table 4 shows a partial version of the results.

Table 4 Experimental results

<i>Number of followers (m)</i>	<i>Leader simulated position (LSP)</i>	<i>Leader estimated position (LEP)</i>	<i> LSP -LEP </i>	<i>Leader simulated direction (LSD)</i>	<i>Leader estimated direction (LED)</i>	<i> LSD -LED </i>	<i>% Error of Direction</i>
1	(27,5)	(26.57,12.69)	7,71	29,31	28,31	1,00	3,41
2	(40,84)	(40.77,83.48)	0,93	125,34	152,58	27,24	21,73
3	(93,39)	(89.36,43.36)	5,68	130,73	128,95	1,78	1,36
...							
8	(25,75)	(24.71,75.64)	0,70	232,81	238,01	5,20	2,23
9	(37,32)	(37.07,31.08)	0,92	189,79	221,80	32,01	16,87
10	(61,67)	(61.76,67.80)	1,10	149,49	157,43	7,94	5,31

Table 4 shows how the number of followers affected the estimation of the position and direction of the leader entity. The results showed a better estimation of both parameters when the number of followers increases. This is reasonable because the more followers there are, the greater the number of regions to intersect and Ω tends to be smaller.

On the other hand, the estimation of the direction did not show in some cases a “good” estimation due possibly to the early alignment of the entities. That is, in the first time-steps, when some estimations were done, the entities just begin to align and ∇ tends to be wider. Even so, the differences between the simulated and calculated data were “small”.

Conclusions and future work

Based on the leadership pattern proposed by Andersson [3], a formal method was proposed to estimate the position and the direction for the leader of a group of entities at a time-point t . The estimations can help to check data consistency, to estimate missing information (an imputation

process) or, simply, to determine where to place and head an entity which is intended to lead the group.

The experiments showed the expediency of the algorithms. Although more extensive experiments are needed, the results suggested that as the number of followers increase, the distance between the estimated and simulated position decreases. With regard to the estimation of the direction, the experiments did not show an analogous behavior; even though most of the times the estimations were very close to the simulated ones.

Next, some directions for future work are considered. To propose alternative methods for estimating the position and the direction of a leader and compare them with the current method. Another interesting issue is to conduct experiments to analyze the effect of parameters such as r , β , and α , that were kept constant here. Experiments with real data are also a must as well as its comparison with Netlogo simulations.

Finally, in most of the flocking systems, followers usually need to know which members are their

leaders. However, other approaches [13-15] have analyzed the behaviour of a leader in flocking systems where the followers (or some of them) do not know which entity is the leader.

Acknowledgements

This paper is a partial result of the project “Optimización de Sistemas de Conocimiento en Ingeniería”, convocatoria nacional para el fortalecimiento de los grupos de investigación y creación artística de la Universidad Nacional de Colombia, 2011-2012

References

1. S. Dodge, R. Weibel, A. Lautenschuz. “Towards a Taxonomy of Movement Patterns”. *Information Visualization*. Vol. 7. 2008. pp. 240–252.
2. M. Benkert, J. Gudmundsson, F. Hübner, T. Wolle. “Reporting Flock Patterns”. *Lecture Notes in Computer Science*. Vol. 4168. 2006. pp. 660-671.
3. M. Andersson, J. Gudmundsson, P. Laube, T. Wolle. *Reporting Leadership Patterns Among Trajectories*. SAC: Symposium on Applied Computing. New York (USA). 2007. pp. 3-7.
4. F. Moreno, E. Ospina. “On Estimating the Position and Direction of a Leader of a Group of Entities”. *Mathematical Modelling in Engineering & Human Behaviour*. Oral communication. Valencia. 2011. pp. 1-2.
5. L. Saul. *Convoy Leader Training: Tactics, Techniques and Procedures*. Center for Army Lessons Learned (CALL) Handbook. U.S. Army. 2003. pp. 1-81.
6. N. Farrington, H. Nguyen, N. Pezeshkian. *Intelligent Behaviors for a Convoy of Indoor Mobile Robots Operating in Unknown Environments*. SPIE: Mobile Robots XVII. Philadelphia. 2004. pp. 26-28.
7. A. Farhangfar, L. Kurgan, W. Pedrycks. “A Novel Framework for Imputation of Missing Values in Databases”. *IEEE Transactions on Systems Man and Cybernetics*. Vol. 37. 2007. pp. 692-709.
8. M. Ian, D. Hoey. *Geometric Intersection Problems*. FOCS: Annual Symposium on Foundations of Computer Science. Houston. 1976. pp. 208-215.
9. C. Murray. *Oracle Spatial. User's Guide and Reference 10g. Release 2 (10.2)*. Ed. Oracle. 2006. pp. 29-46.
10. U. Wilensky. *NetLogo Flocking Model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Disponible en: <http://ccl.northwestern.edu/netlogo/models/Flocking>. Consultado en Febrero 5 de 2011.
11. S. Momen, B. Amavasai, N. Siddique. *Mixed Species Flocking for Heterogeneous Robotic Swarms*. IEEE Eurocon: The International Conference on Computer as a Tool. Warsaw. 2007. pp. 2329 - 2336.
12. F. Stonedahl, U. Wilensky. *Finding Forms of Flocking: Evolutionary Search in ABM Parameter-Spaces*. AAMAS: International Conference on Autonomous Agents and Multi-Agent Systems. Toronto. 2011. pp. 61-75.
13. W. Zongyao, G. Dongbing. *Distributed Cohesion Control for Leader-follower Flocking*. FUZZ-IEEE: IEEE International Conference on Fuzzy Systems. London. 2007. pp. 23-26.
14. S. Housheng, W. Xiaofan, L. Zongli. *Flocking of Multi-agents with a Virtual Leader Part I: With a Minority of Informed Agents*. IEEE Conference on Decision and Control. New Orleans. 2007. pp. 2937-2942.
15. H. Su, G. Chen, X. Wang, Z. Lin. *Adaptive Flocking with a Virtual Leader of Multiple Agents Governed by Nonlinear Dynamics*. CCC: Chinese Control Conference. Beijing. 2010. pp. 29-31.