Simulation of the Flocking Behavior of Birds with the Boids Algorithm

CARL-OSCAR ERNEHOLM

KTH Computer Science and Communication

Bachelor of Science Thesis
Stockholm, Sweden 2011

Simulation of the Flocking Behavior of Birds with the Boids Algorithm

C A R L - O S C A R E R N E H O L M

Bachelor's Thesis in Computer Science (15 ECTS credits)
at the School of Computer Science and Engineering
Royal Institute of Technology year 2011
Supervisor at CSC was Lars Kjelldahl
Examiner was Mads Dam

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/
erneholm_carl-oscar_K11044.pdf

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.kth.se/csc

Abstract

Few things in nature is as impressive as how some animals seems to be able to organize themselves into larger groups so effortlessly. By learning more about this fascinating behaviour we might be able to apply this knowledge to find new solutions to our own problems.

One of the first stepping stones to get to an understanding of flocking behaviour is to be able to simulate it. The first flocking-behavior simulation was done on a computer by Craig Reynolds in 1986 he called his simulation program: "Boids". It's still to this day to most used model for simulating flocking behaviour.

This thesis attempts to compare two different definitions of the neighbourhood of the boids. The definition of the neighbourhood has great effect of how the boids act.

Referat

Simulering av flockbeteendet hos fåglar

Få fenomen i naturen är så häpnadsväckande som flockande fåglar och stimmande fiskar. Från djurs beteenden kan vi hämta inspiration till nya lösningar på problem vi står inför.

Första steget att förstå djurs flockbeteende är att hitta en modell som kan simulera beteendet. 1986 konstruerade Craig Reynolds den första lyckade modellen för simulering av flockbeteende, han implementerade den som ett datorprogram som kallades "Boids". Modellen som Boids är baserad på är än idag den mest använda.

Detta kandidatexamensarbete ämnar att jämföra två olika möjliga definitioner av en boids "grannskap" (neighbourhood), grannskapets definition har stor inverkan på hur Boids modellen beteer sig.

Contents

Contents

Chapter 1

introduction

1.1 Background

Animal behavior has always been a source of amazement to mankind. In many areas the abilities of the animals surpasses the abilities of us humans, but with the use of technology we have been able to best the animals in more and more areas. By studying the behavior of animals we have been able to find many interesting solutions animals apply to the problems in their environment. These solutions has inspired many people to think of problems in new ways, sometimes leading to new solutions, often with great results. But few things are more impressive than the way some creatures organize themselves in larger groups: birds, fishes and ants; flocks, shoals and swarms. If we can understand the mechanism of how these organizations emerge, we might be able to use the same mechanics to achieve what we have not achieved before.

One of the first stepping stones to get to an understanding of flocking behaviour[1] is to be able to simulate it. With a simulation we could easily play around with the different parameters and find a variant that suits our needs.

The first flocking-behavior simulation was done on a computer by Craig Reynolds in 1986 he called his simulation program: "Boids"[2]. Boids was named after the simple agents in the system which he also called boids. The boids used a set of simple rules to determine how they would move. The three rules formulated by Reynolds in his Boids program are still the basis of modern flocking simulation and are widely used to these days. These are the three rules as they are described by Reynolds on his website [3]:

[1] I will use the term flocking when referring to all sorts of flock-like behavior. Much of the mechanisms that will be described can be applied to other species than birds.

Separation Steer to avoid crowding local flockmates.

Alignment Steer towards the average heading of local flockmates.

Cohesion Steer to move toward the average position of local flockmates.

In 1990 Frank Heppner and Ulf Grenander proposed another model for simulating flocking behavior [1]. Their model also consists largely by the application of three rules:

Homing each member of the flock tries to stay in the roosting area.

Velocity regulation each member of the flock tries to fly with a certain predefined flight speed - it tries to return to that speed if perturbed.

Interaction if two flockmates are too close to one another, they try to move apart; if they are too distant, they do not influence each other; otherwise they try to move closer together.

One of the primary features of this model (in contrast to Reynolds model) is the inclusion of random disturbances[2]. It simulated this disturbances with a Poisson stochastic process, however one of the weaknesses of this model is that it won't yield satisfiable results without these disturbances.

In this report I decided to focus on Reynolds model for flocking behavior, for several reasons:

- It is the most widely used of the two.
- There is a lot more material concerning Reynolds model.
- It seems to be the most simple and elegant of the two as it does not rely on external disturbances.

Also for simplicity I will limit myself to look at the model as it applies to simulating flocking behavior on a 2D plane, however the concepts described in this report could easily be translated to more dimensions.

[2]This models sudden gusts of wind, or other distractions that would influence the course of the boids.

1.2 Problem

While the idea and structure of Reynolds Boids algorithm is quite clear cut, it leaves some room for tweaking and modification to achieve even more convincing results. This thesis attempts to compare two ways of defining the neighbourhood of the Boids in the Boids algorithm, to find which yields the best results. The first is the original neighbourhood as used in the Boids implementation: all boids within a certain radius are in the neighbourhood. The other definition is the N closest boids are the neighbourhood.

Intended Behaviour

But how should a flock behave? There's no formal definition of how a group of animals should behave to be a "flock". For the sake of the comparison, we will define flocking as a group of moving bodies that:

- Are close to each other.
- Move with approximately the same speed.
- Move in approximately the same direction.

And "good" flocking behaviour is defined as:

- Bodies that meet should form a flock.
- The flock member should not collide with each other.
- The flock should not spontaneously lose members.
- The flock should not spontaneously split off into new flocks.

1.3 Method

To test these two definitions of the neighbourhood I will implement my own version of the Boids algorithm. In this way I will have full control of the implementation so that I can make any changes I need.

Both of the definitions will be tested against the criteria defined in section: 1.2. Each variant will run through a predefined set of tests. They will also be visualized with both position and heading of each boid will shown. In this way any significant differences between the different approaches should be easy to spot.

The result and discussion part will then make an effort to explain why the differences look like they do and try to find the most prominent strengths and weaknesses of both approaches.

1.3.1 The Test Cases

Since it's impossible to test all the possible situations the boids could find themselves in, we have to chose carefully what situations we want to test. These are the situations the test cases should aim to simulate.

1. A single boid meets a single boid.
2. A few single boids meet.
3. A single boid meets a group of boids.
4. Two boid flocks of equal size meet.
5. Two boid flocks of unequal size meet.

There is 15 test cases in total, each goal on the list has three tests with varying flock sizes and initial relative heading (where applicable). For a few examples of the test cases used see appendix A. For more details on the test cases please refer to the implementation.

1.4 Definitions

1.4.1 Boid

A representation of a bird in flocking simulations. The word is most commonly used in the context of Boids, the original computer simulation of flocking behavior by Reynolds and derivatives of Boids.

1.4.2 Flocking Behavior

In this report flocking behaviour is used very lousily to mean any flock-like behaviour, such as: Shoaling, swarming and maybe to some extent human crowds.

1.4.3 Emergent behaviour

When individual objects interact with each other directly or indirectly to create much more complex results. Another characteristic of emergent behaviour is that the end result is hard to predict even if each interaction in itself is simplistic. Ant hills, economics, the evolution are all examples of how agents abiding to relatively simple rules can create a system more complex than the sum of its parts. Reynolds Boids algorithm can be said to create an emergent behaviour.

Chapter 2

Basic Concepts

2.1 Reynolds Boid Model and Visualisations

Reynolds boid model follows like most other visualisations this simple structure: calculate the current state, draw the state on the screen and repeat. Algorithm 1 shows the structure of a typical Boids implementation, the first of the innermost for loops is the calculation step. The three rules of the Boids algorithm only changes the speed and course of the boid and can be applied in any order. After the speed and course of the boids has been updated by Reynolds rules we can update the positions of the boids. We update the position by simply calculating where it would be if it we let it fly straight ahead for some small amount of time (given the new speed and course).

After all the boids has been updated with new positions, speed and course we draw the current state on the screen and repeat.

Data: A group of boids.
Result: Simulates flocking behaviour with an animation.

```
foreach Frame do
    foreach boid do
        separation(boid);
        cohesion(boid);
        alignment(boid);
    end
    foreach boid do
        boid.x ← cos(boid.course) * b.velocity * dTime;
        boid.y ← sin(boid.course) * b.velocity * dTime;
        draw(boid);
    end
end
```

Algorithm 1: An overview of the Boids algorithm.

2.2 The Boid

The boid is the bird representation in Reynolds flocking simulation model. Each boid object should at least have the following attributes to describe the state it is in.

location	The x and y coordinates of the current position of the boid.
course	The angle of the current course of the boid.
velocity	The speed of which the boid is traveling.

The course and speed could of course be represented by a equivalent velocity vector instead. But often more attributes is needed to make the simulation more convincing. Putting an upper limit on how fast the boids can move and turn is a common improvement.

2.3 Cohesion

Data: A boid.
Result: The course of the boid is updated.

goal ← (0,0);
neighbours ← getNeighbours(boid);
foreach *nBoid in neighbours* **do**
  goal ← goal + positionOf(nBoid);
end
goal ← goal / neighbours.size();
steerThoward(goal, boid);

Algorithm 2: Reynolds first rule: Cohesion.

Steer to move toward the average position of local flockmates.[3] Cohesion is the rule that keeps the flock together, without it there would not be any flocking at all. Algorithm 2 shows how this algorithm could be implemented, it finds the average position of the neighbourhood boids and tries to move the boid towards it.

2.4 Separation

Data: A boid.
Result: The course of the boid is updated.

goal ← (0,0);
neighbours ← getNeighbours(boid);
foreach *nBoid in neighbours* **do**
  goal ← goal + positionOf(boid) - positionOf(nBoid);
end
goal ← goal / neighbours.size();
steerThoward(goal, boid);

Algorithm 3: Reynolds second rule: Separation.

Steer to avoid crowding local flockmates.[3] If a flocking behaviour is to be convincing it must also avoid collisions between the boids. This rule attempts steer the boid away from possible collisions. It's important to note that the distance from which the boids start to avoid each other must be less than the distance from which the boids attract each other (due to the cohesion rule). Otherwise no flocks would be formed.

2.5 Alignment

```
Data: A boid.
Result: The course and velocity of the boid is updated.
dCourse ← 0;
dVelocity ← 0;
neighbours ← getNeighbours(boid);
foreach nBoid in neighbours do
    dCourse ← dCourse + getCourse(nBoid) - getCourse(boid);
    dVelocity ← dVelocity + getVelocity(nBoid) - getVelocity(boid);
end
dCourse ← dCourse / neighbours.size();
dVelocity ← dVelocity / neighbours.size();
boid.addCourse(dCourse);
boid.addVelocity(dVelocity);
```

Algorithm 4: Reynolds third rule: Alignment

steer towards the average heading of local flockmates.[3] This rule tries to make the boids mimic each others course and speed. If this rule was not used the boids would bounce around a lot and not form the beautiful flocking patterns that can be seen in real flocks.

2.6 Defining the Neighbourhood

The neighbourhood of a boid is the neighbourhood it perceives and is a core part of Raynolds formula. The neighbourhood decides what other boids a boid should take into account when deciding the next move.

Each pseudo code of the three rules of the Boids algorithm (algorithms 2, 3 and 4) uses a function called getNeighbours(Boid) to get the neighbors of the given boid. The three rules could and should use different implementation of the getNeighbours(Boid) method. It's important for example that the separation rule only acts on the closest boids. If separation and cohesion would work on the same neighbourhood they would cancel each other out.

Raynolds original implementation simply defined the neighbors as the boids within a certain radius. Another possible definition of the neighbourhood is to let each boid look at the N closest boids instead. This thesis attempts to compare the two definitions to find strengths and weaknesses in both definitions and find under what circumstances either is superior.

Chapter 3

Result and Discussion

All the test cases was performed to compare the original boids definition of the neighbourhood[1] with the N closest boids. Several values of N was tested but all values had similar problems. The biggest problem with using the N closest as neighbourhood function in both the alignment and cohesion part of the algorithm is that both rules get activated at the same time. Each time a boid got attracted to a boid by the cohesion rule the alignment rule would hinder it.

Test case no. 8 is a good example of this, see Figure A.1 in appendix A. Notice how the single blue boid in the upper left corner starts off by steering away from the blue flock. It's the alignment rule that produces this behaviour. The boids in the flock is the closest boids known to the single blue boid. The single red boid won't have have either the cohesion or alignment act upon it, until the red flock is close by.

This problem could be resolved by weighting the cohesion to be more effective at a longer range and make alignment more effective at shorter range. However it's not easy to get this weighting right.

Another solution would be to only allow a boid to be active in one of the rule for each boid at the time. If the boids are too close the separation rule would be used to avoid a collision. If the boids has a near optimal range from each other (within flocking distance, but outside collision distance) the alignment rule would act upon them, too keep them in this good state. If the boids are far from each other the cohesion rule would bring them together within flocking distance. However the testing of this neighbourhood definition is however outside of the scope of this thesis.

But why can't we just use the original neighbourhood definition with the in-

[1] All boids within a certain radius of the boid is the neighbourhood.

creased range? Why must we only look at the N closest boids? If a boid takes influence from too many boids, the cohesion and alignment rule will be ineffective. This is because these rules uses mean values of the neighbourhood; with more boids the risk increases that the various influence will cancel each other out. It's therefore a good idea to set a limit to the number of boids in the neighbourhood.

But the neighbourhood definition of the original Boids algorithm is not without it's flaws. The biggest problem the original faces is that since the boids cannot have too many neighbors, we have to limit this the radius from within the boid senses it's neighbors. We are therefore forced to make the radius short. But with a short radius of neighbors a single boid won't find flockmates unless they bump into each other. Boids using the within radius neighbourhood definition risk flying past each other relatively close, this is not good flocking behaviour.

However even the N closest boids has this issue, when they are in a flock of size greater or equal to N. When a flock is of size N they won't be effected by other boids unless they get closer than the rest of the flock.

This problem that both definitions faces can clearly be seen in test case number 11 (see figure A.2 in appendix A).

Since both of them got the same problem for different reasons, it might be possible to combine them and use the strength of both to avoid the problem. What if define the neighbourhood as both the N closest outside the neighbourhood radius *and* all boids within the smaller radius? Maybe it could be possible to weight it so that the within radius neighbourhood keeps the boid close to the neighbourhood, while the N nearest would attract new boids to the group. However, again this topic lays outside the scope of this thesis.

Chapter 4

Conclusion

The definition of the neighbourhood is an interesting topic. While the definition of the neighbourhood does not in any way change the classic rules formulated by Reynolds, it does profoundly change the behaviour of the simulation. This thesis compared the Reynolds classical Neighbourhood definition: all boids within a certain radius is the neighbourhood, with the N closest boids are the neighbourhood.

The advantage of N closest definition is that it allows a larger search radius since the number of neighbors is not limited by the radius but of N. If one choses to use N closest it's important to lower the weight of the alignment if the boid is far away or else the boids will steer away from each other instead of forming a flock. However if the all boids has formed flocks of sizes greater than N it becomes harder for each individual flock to notice the other flocks. Since the boids in each flock only sees the N closest boids within the flock, it won't see the other flocks unless they bump into each other.

This is less of an issue with the within a radius definition, since it will always notice boids within the radius. Thus I would recommend the within a certain radius definition if the system contains many boids. But if the boids is separated by a great distance the N closest would be a good choice since it's good at finding new boids until it's in a flock of size greater than or equal to N.

In the end both definitions has its strengths and weaknesses, what definition is best depend on the circumstances.

Bibliography

[1] F. Heppner and U. Grenander. A stochastic nonlinear model for coordinated bird flocks. 1990.

[2] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics*, pages 25–34, 1987.

[3] Craig W. Reynolds. Boids, Background and Update . `http://www.red3d.com/cwr/boids/`, 2007. [Online; accessed 14-April-2011].

Appendix A

Illustrated Test Cases

The blue boids in the figures use the N Closest definition while the red ones use the original all boids within a radius definition of the neighbourhood. The blue and red boids do not interact with each other, only boids of the same color interact. In all of the examples N is set to 7.

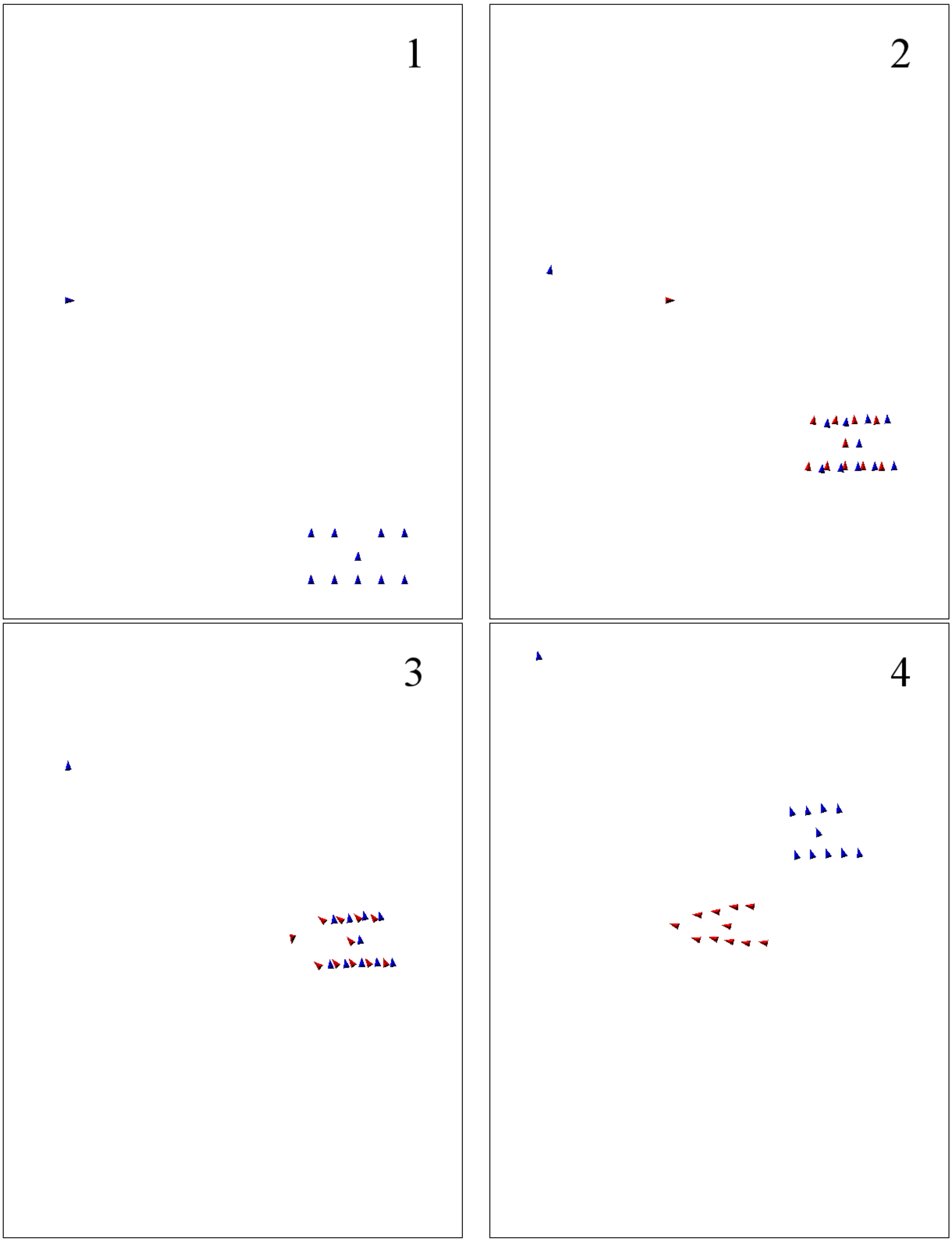

Figure A.1. Test case 8.

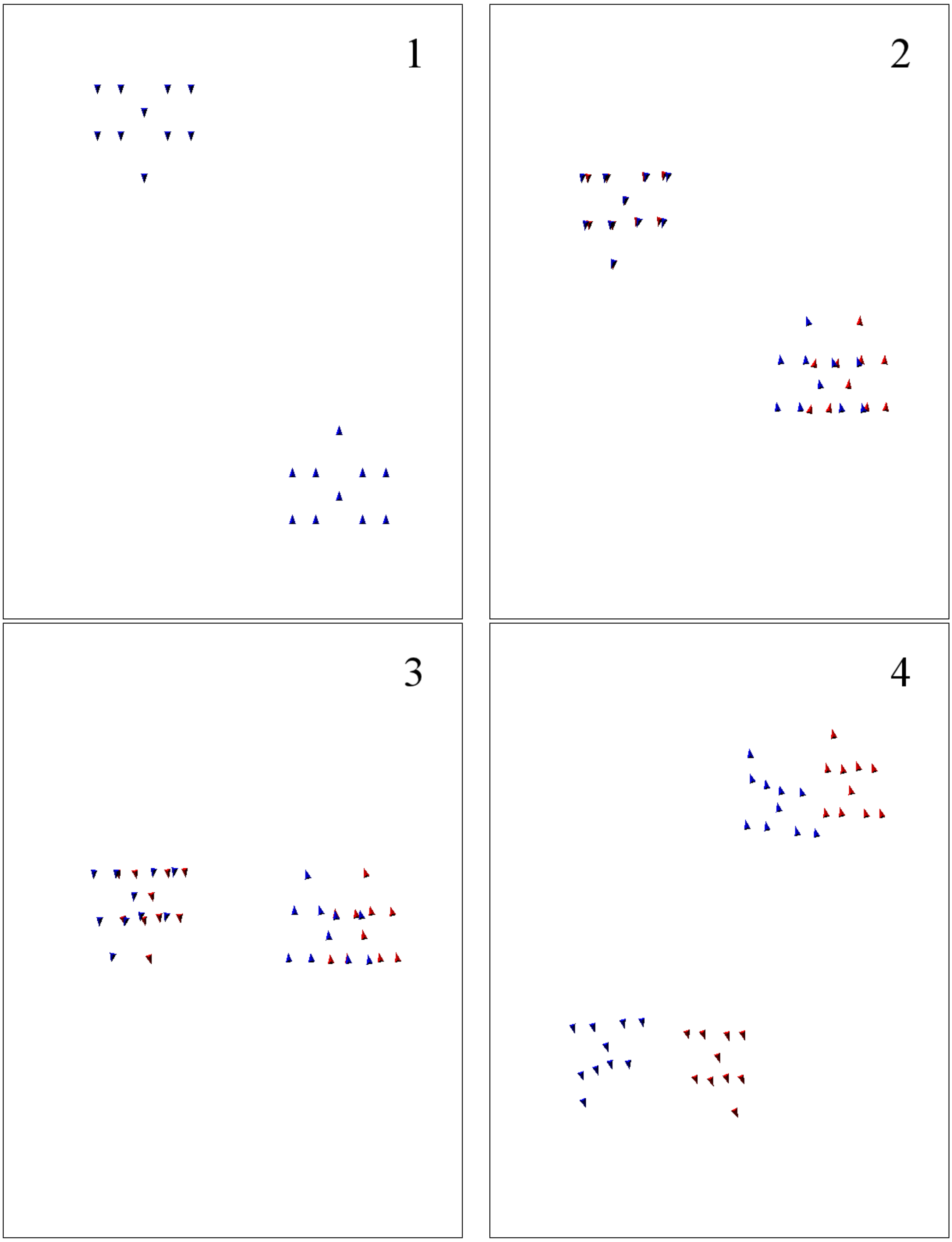

Figure A.2. Test case 11.