International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-2, Issue-2, February 2014

Auxiliary Steering Behaviour for Computer Based Intelligent Agents

Bhagya Shree Jain, Sagar Chandrakar and Shrikant Tiwari

Abstract— **This paper is the next evolutionary step towards steering behaviours through intelligent agents populating virtual or simulated environments. We are now presenting additional or auxiliary steering behaviour which will present a more reliable approach towards automatic path planning of intelligent computer agents and would focus more on group behaviour of intelligent agents. This paper intent to increase the level of accuracy as well as reliability of autonomous locomotion among the group by calculating and acting according to the location of other intelligent agents in such a way that optimum solution or way is available at all times. In this papers different group steering ways are examined. Our approach can be adapted to any place where group behaviour in locomotion is desired or required.**

***Index Terms*— Virtual environments, intelligent agent, steering behaviour.**

I. Introduction

Solutions for manyrequirements of intelligent agents in animation, games, virtual reality and robotic movement are solved by the steering behaviour.These Steering Behaviours are more or less independent of the particulars of the agent's way of movement. They provide the ability to navigate around domain in a life-like and improvisational manner. Combinations of steering behaviours can be used to much higher degree of freedom in locomotion. This paper presents efforts with the goal of allowing a control of huge number of intelligent agents in an interactive virtual environment. This work isspecially oriented to the control of intelligent agents for their movement for simulation experiments[1, 2].This approach can be tailored to suit any particular needs may be for training robots or for real life applications.

To cater the need of computer animation, interactive media such as games and virtual reality there is need for intelligent agent, those who can navigate according to requirement. These agents may have some ability to improvise [3] their actions according to situation. This paper will use the term intelligent to denote these agents which are intelligent agents in a virtual world who may be interacting [4] with other agents. In this paper the term behaviour is used to refer to the improvisational and life likes actions of an intelligent agents. This paper will focus on steering, the middle layer of the behavioural hierarchy [5],it will briefly describe a simple model of the locomotion layer [6], but only in just details to provide a solid foundation for the discussion of various steering behaviours, movements [7] and functions.

Manuscript received Feb. 20, 2014.

Bhagya Shree Jain, Department of Computer Science and Engineering, FET, SSTC, SSGI, Junwani, Bhilai (C.G.)

Sagar Chandrakar, Department of Computer Science and Engineering, FET, SSTC, SSGI, Junwani, Bhilai (C.G.)

Shrikant Tiwari, Department of Computer Science and Engineering, FET, SSTC, SSGI, Junwani, Bhilai (C.G.)

II. Environment and Intelligent Agent

Computer simulation environments are composed of large amount of different agents where each one has its own unique behaviour and all of them are interacting and competing with each other. The objective of constructing simulation scenarios within such environment is one of the most time and effort consuming tasks. The possibility to include intelligent elements that react with each other using its internal defined behaviour can dramatically simplify the creation of realistic scenarios. Agents or elements must be guided by lines of code, consisting of sequence of commands. Depending upon the requirement or experiment specification, changes can be done in destination, behaviour , acceleration, speed or manipulation of the intelligent agent. The approach taken in this paper is to consider steering behaviours as essentially independent from the underlying movement or navigation stream.

This locomotion model will be based on a simple idealized vehicle concept. This vehicle model is based on a point mass approximation [8]. On the one hand it allows a very simple and computationally cheap physically-based model. On the other hand, it cannot be a very complex physical model because point masses do not exist in the real world. Any physical object with mass must have a positive radius and hence a moment of inertia.

A point mass is defined by a variable of position property and a mass property. In addition, it has velocity property. The velocity can be modified by applying forces. Since it is a vehicle, these forces are generally self-applied, and hence limited. Let's take the case where there is a typical force which adjusts a vehicles velocity is thrust, generated by the vehicles own power source, and hence limited in magnitude by the capacity of the power source. For the simple vehicle model, this notion is summarized by a single Maximum force. Most vehicles are characterized by a maximum speed. Largely this limitation is due to the interaction between acceleration due to their finite thrust and the deceleration due to various factors. As an alternative to realistic simulation of all these limiting forces, the simple vehicle model includes a Maximum Speed variable. This speed limit is enforced by a dynamics of the vehicles velocity vector. Finally, the simple vehicle model includes an orientation, which when taken together with the vehicles position form a velocity-aligned local coordinate space to which a geometric model of the vehicle can be attached.

In this model, the control signal passed from the steering behaviours to the locomotion behaviour may consist of exactly one vector quantity which is a desired steering force. More realistic vehicle models can have very different sets of control signals. A generalized control system can be described as an agent that uses the information provided from the input set to perform the required transitions of state and to evaluate the output information which are sent to the controlled system. It is possible to map a generic steering force vector into these scalar signals where the side component of the steering vector can be interpreted as the steering signal; the forward component of the steering vector can be mapped into the accelerator and brake according to their sign magnitude. Another feature of this model is that it allows the vehicle to turn when its speed is zero. Most real vehicles cannot do this. This problem can be solved by placing a few additional constraints on change of orientation, or by limiting the lateral steering component at low speeds, or by simulating moment of inertia. Thus for intelligent agents there movement, conceptualization, visualization and application depend upon the choice of the software or simulating environment [9].

III. Auxiliary Steering Behaviour

Now in context of steering behaviours lets assumes that there navigation is implemented by the simple vehicle model described above, and is parameterized by a single steering force vector. Therefore the steering behaviours are now described in terms of the geometric calculation of a vector.
The first in sequence is Path Following [8]. This behaviour enables an intelligent agent to steer along a predetermined path or way as shown in Figure 1. This is a deviation from constraining a vehicle rigidly to a path like a car moving along a track. Rather path following behaviour is desired to produce motions such as cars moving down an express way: the individual paths remain near, and often parallel to, the centreline of the corridor, but are free to deviate from it. In the implementation described below, a path will be visualized as a spine and a radius. The spine might be represented as a spline curve. The path is then a tunnel or generalized cylinder, a circle of the specified radius, swept along the specified spine. The goal of the path following steering behaviour is to move intelligent agent along the path while remaining within the specified radius.

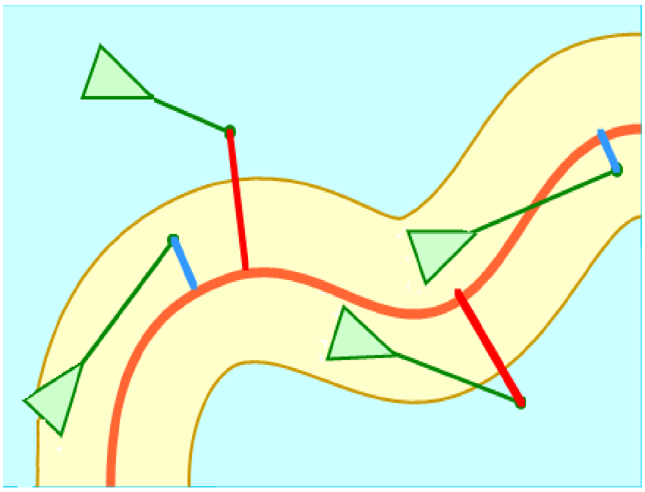

Figure 1: Path Following [8]

Path Following

Desired Velocity = (Current Position – Target Position) * Maximum Speed

If (Steering Force Is Positive)

Steering Force = Desired Velocity – Current Velocity

Else

Steering Force = Current Velocity - Desired Velocity

Formula Set 1: Path Following

The details of calculations for path following is mentioned in Formula set 1.The next three steering behaviours: separation, cohesion, and alignment, relate to group behaviour. In each one of them, the steering behaviour determines how an agent reacts to other agent in its local neighbourhood. Intelligent agent outside domain or range is ignored. As shown in Figure 2, the neighbourhood is specified by a distance which defines when two agents are nearby, and an angle which defines the agent's perceptual field of view.

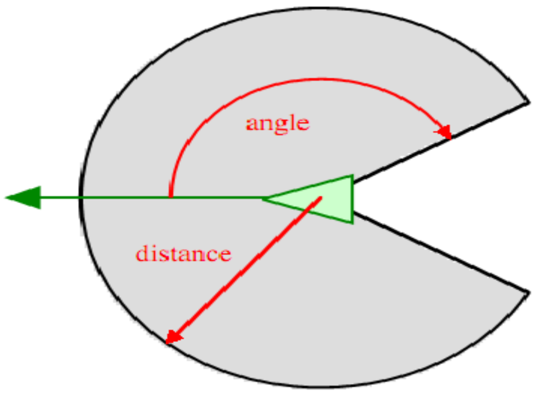

Figure2: Neighbourhood [8]

Separation

Distance = Target Position – Current Position

Desired Velocity = (Current Position – Target Position) * Maximum Speed

Steering Force = Desired Velocity – Current Velocity

Cohesion

Distance = Target Position – Current Position

Desired Velocity = (Current Position – Target Position) * Maximum Speed

Steering Force = Current Velocity - Desired Velocity

Alignment

Distance = Target Position – Current Position

Desired Velocity = (Summation of all(Current Position – Target Position))* (Maximum Speed / Number)

Steering Force = Desired Velocity – Current Velocity

Formula Set 2: Separation, Cohesion, Alignment

Separation steering behaviour [8] gives the intelligent agents the ability to maintain a certain separation distance from others nearby agents (See Figure 3). This is used to prevent agents from crowding together. To compute steering for separation, first a search is made to find other agents within the specified range or domain area. This might be an exhaustive search of all agents in the simulated world, or may use spatial partitioning or caching scheme to limit the search to local agents. For each nearby agent, a repulsive force is computed by subtracting the value of positions of our agent with the nearby agent, then performing normalizing and then applying a reduced weight. These repulsive forces for each nearby agent are clubbed together to produce the overall steering force.

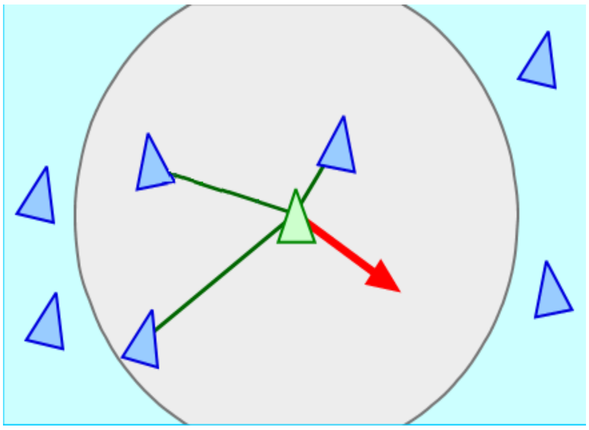

Figure 3: Separation [8]

Cohesion steering behaviour [8] gives an agent the ability to approach and form a group with other nearby agents (Figure 4). Steering for cohesion can be computed by finding all agents in the local neighbourhood, computing the average position of all the nearby agents. The force can applied in the direction of that average position by subtracting our agent position from the average position.

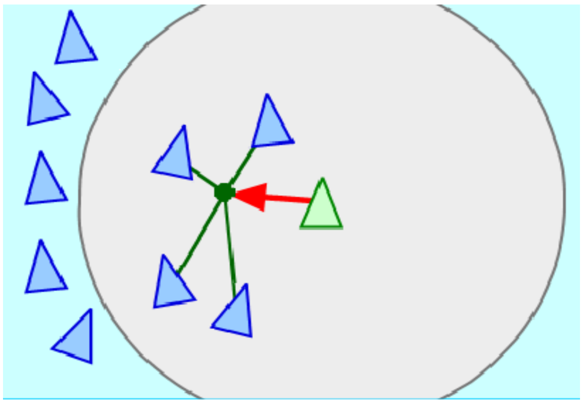

Figure 4: Cohesion [8]

Formula Set 2 shows the calculation used behind group behaviour.

Alignment steering behaviour [8] gives an agent the ability to align itself with other nearby agents, as shown in (Figure 5).Steering for alignment can be calculated by finding all agents in the local neighbourhood and then averaging together the velocityof the nearby agents. This value is the desired velocity, so the steering vector is the difference between the average and our agent's current velocity. This steering will tend to turn our agent.

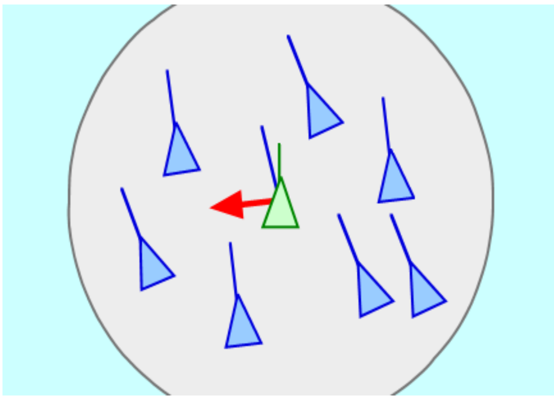

Figure5: Alignment [8]

IV. Observation

There are several factors which affect the performance of various steering behaviour, relationship is shown in Figure 6. For auxiliary steering behaviour which is more inclined towards the group behaviour the major factors are speed, mass, number of other intelligent agents as well as the behaviour and position of other intelligent agents as they follow group dynamics.

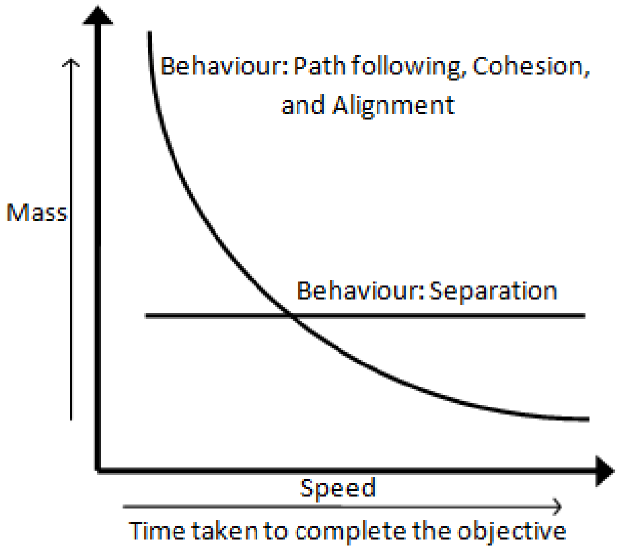

Figure 6: Correlation of various factors of intelligent agents in different behaviour.

The Table 1 below is the tabular representation of desired qualities of intelligent agents for optimum performance.Since we are following group dynamics the individual properties as well as group properties plays a very important role in the outcome of any locomotion. Performance also depends upon constrains placed upon by the simulating environment. The environment has huge role defining various values which are thus reflected in individual agents thus altering the outcome.

Behaviour	Desired Qualities
Path Following	More Mass, Less Speed
Cohesion	More Mass, Less Speed
Alignment	More Mass, Less Speed
Separation	Depends upon number of other agent

Table 1: Desired qualities in intelligent agent for better performance.

Figure 7 provides us with the relationship between the mass and speed of individual intelligent agents as these two qualities are most important in determining the behaviour of the agents for optimum performance.

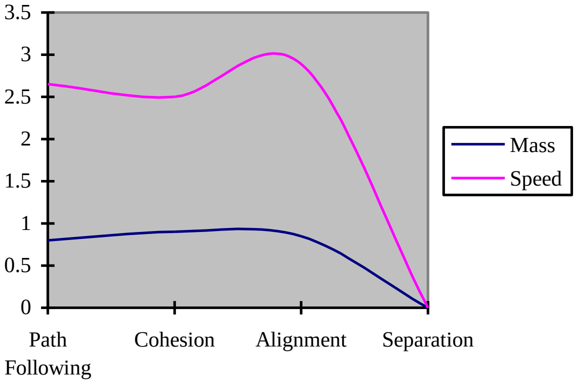

Figure 7: Correlation of Speed and mass for optimum performance

For the separation behaviour the mass and speed has practically no implications for rest both are very important part in determination of speed and direction in locomotion.

V. Conclusion

This paper first conceptualized a simple point mass agents where its locomotion was defined by auxiliary steering behaviour of intelligent agents for improvisational action and path following. Then we observed various factors which affect agents and various their behaviour aspects when combined with steering behaviour. Then various group steering behaviour were examined and there approach towards locomotion of intelligent agents was viewed.

References

[1] J. Miguel Leitao, A. Augusto Sousa and F. Nunes Ferreira, "Graphical Control of Autonomous, Virtual Vehicles", in Proceeding of IEEE Vehicular Technology Conference, pages 507-511. May 2000.

[2] Hongling Wang, Joseph K.Kearney, James Cremer and Peter Willemsen, "Steering behaviors for autonomous vehicles in virtual environments", in Proceeding of IEEE Virtual Reality Conference, pages 155-162.March 2005.

[3] Alejandro Moreno Astorga, David Moreno-Salinas, Dictino Chaos García and Joaquín Aranda Almansa, "Simulation Benchmark for Autonomous Marine Vehicles in LabView" in Proceeding of OCEANS Conference, pages 1-6. June2011.

[4] D.K. Liu, X. Wu, A. K. Kulatunga and G. Dissanayake, "Motion Coordination of Multiple Autonomous Vehicles in Dynamic and Strictly Constrained Environments" in Proceeding of IEEE Cybernetics and Intelligent Systems Conference, pages 1-6. June 2006.

[5] Bruce M. Blumberg and Tinsley A. Galyean, "Multi-Level Direction of Autonomous Creature for Real-Time Virtual Environments" in Proceeding of SIGGRAPH 95 Conference, pages 47-54. August 1995.

[6] Sergio Reis Cunha, Anibal Coimbra de Matos and Fernando Lobo Pereira, "An Automatic Path Planing System for Autonomous Robotic Vehicles" in Proceeding of IECON 93 Conference, vol.3, pages 1442-1447. November 1993.

[7] Faiz BenAmar, "Steering behaviour and control of fast wheeled robots" in Proceeding of IROS 97 Conference, vol.3, pages 1396-1401. September 1997.

[8] Craig W. Reynolds, "Steering Behaviors for Autonomous Characters" in Proceeding of Game Developers Conference, pages 763-782. May 1999.

[9] Lyle N. Long, Scott D. Hanford, Oranuj Janrathitikarn, Greg L. Sinsley and Jodi A. Miller, "A Review of Intelligent Systems Software for Autonomous Vehicles" in Proceeding of Computational Intelligence in Security and Defence Applications Conference, pages 69-76. April 2007.