2012 IACSIT Hong Kong Conferences
IPCSIT vol. 30 (2012) © (2012) IACSIT Press, Singapore

Simulation of Flocking Behavior in Game by Human Emotions – Using Embedded Support Vector Machine

Jung-Ying Wang [1 +] and Wei-Han Li [2]

[1] Department of Multimedia and Game Science, Lunghwa University of Science and Technology, Taiwan
[2] Department of Electronic Engineering, Lunghwa University of Science and Technology, Taiwan

Abstract. In this paper, we select the appropriate items from human emotions to implement real time flocking behavior in game. We first use two popular machine learning methods, neural network (NN) and support vector machine (SVM), to predict which emotion should be excited for a non-player character. The experiment results show that the applications for both neural network and support vector machine in 3-fold cross-validation achieve the prediction rate of 95.61% and 99.96%, respectively. In other words, both of them can successful using as the on-line predictor of excited emotion in game, and can easily replace the traditional method of finite state machine. After that, we develop a simulation system which embedded above trained SVM module and Reynolds flocking rules to simulate the flocking behavior by human emotions. The experiment results show that when a group of non-player characters is activated by a special type of emotion, e.g. angry, our simulation system can successfully simulate the non-player characters all having the human-like (or animal-like) behavior.

Keywords: flocking, human emotions, game artificial intelligence, non-player character.

1. Introduction

The next revolution in games is not technological – it is emotional [1]. Therefore, how to create or use emotions in games is an import issue in these days. Olsen et al. [2] demonstrate that adding emotions to a real-time system with artificial intelligence is feasible and warrants further study. Bernhaupt et al. [3] developed a game called "Emotional Flowers". This game enables the player to simply grow a plant based on their emotions. Aqel et al. [4] describes a simulation model for analyzing artificial emotion injected to design the game characters. But it is very seldom to try to simulate non-player characters flocking behavior in game by considering emotions. In this paper, we develop a simulation system which embedded trained SVM module as excited emotion predictor and Reynolds flocking rules to simulate the flocking behavior of non-player characters by human emotions.

For human emotions, some contemporary emotion theorists have proposed some basic emotions. For example, Tomkins [5] has proposed eight basic emotions, including anger, interest, contempt, disgust, distress, fear, joy, shame and surprise. However, there is little agreement about how many emotions are basic and which emotions are basic [6]. Ortony et al. [6] give a more complete overview about contemporary emotion theorists. In order to have a more rational mapping from a human emotion to a non-player character of role-playing game. The emotions of anger, distress, fear and joy are selected as basic emotions, and we utilize them as the feature vectors in our game's simulation.

2. Methods

2.1. Data sets and coding scheme

+ Corresponding author. Tel.: +886-921241960; fax: +886-2-82094650.
E-mail address: *wyy@mail.lhu.edu.tw*.

Two machine learning methods, the neural networks and support vector machine, are used as predictor to predict which emotion will be excited. In order to develop our prediction model, four basic motions anger, distress, fear and joy are selected as the feature vectors. All emotions exist on a scale of 0 to 100. In this study, the coding scheme uses the method of exhaustion (brute force method) to code. That is, the score range of each basic emotion is set from 0 to 100 increments of 10 with 0 the lowest and 100 the highest. We randomly selected 2/3 of the data as training data, the remaining 1/3 of the data as the test data. Based on this coding, there will be 11*11*11*11 = 14641 possible patterns in our data set. Because of the emotion simulation in game, only the strongest emotion will be excited at each time frame. Therefore, we take the highest score of the four basic emotions as the excited state. For example, a non-player character datum with emotion score of anger = 30, distress = 50, fear = 70 and joy = 40, then the basic emotion of fear will be excited. But if the highest score of any two or more of the four basic emotions are the same, then we set its state to "undetermined", and random choose one of the highest emotion score as the excited state. For example, a non-player character datum with score of anger = 10, distress = 50, fear = 50 and joy = 30, then the state will be set to undetermined and the excited state will random select from the emotion of distress or fear. In other words, each data is set to one of four emotions state or set to the "undetermined" state.

2.2. Software and model selection for NN and SVM

In our study, a computer program was implemented using WEKA 3.6 [7] for the back-propagation neural network model. There were four input nodes and five output node. The maximum iteration for the back-propagation neural network was set as 500. The learning rate and momentum were set to as 0.3 and 0.05, respectively. It is worth mentioning that there is no standard method to get the number of nodes needed in the hidden layer. In here, 20 nodes were used in the hidden layer by try and error.

We use the software LIBSVM 3.1 [8] for experiments. LIBSVM is a general library for support vector classification and regression. There is different functions φ to map data to higher dimensional spaces; practically we need to select the kernel function. There are several types of kernels in used with all kinds of problems. Each kernel may be more suitable for some problems. In our experience, the RBF kernel is a decent choice for most problems. A learner with the RBF kernel usually performs no worse than others do, in terms of the generalization ability. In this paper, we did some simple comparisons and observed that using the RBF kernel the performance is better than the linear and polynomial kernel for the problem we studied. We then use the RBF kernel for all the experiments.

Another important issue is the selection of parameters. For SVM training, few parameters such as the penalty parameter C and the kernel parameter γ of the RBF function must be determined in advance. Choosing optimal parameters for support vector machines is an important step in SVM design. We carried out some experiments to turn up and evaluate the prediction system by random selecting about 1/n data as training data and testing the selected model by another 1/n, where n equal to seven for our data set. After this we could find the optimal pairs of C and γ for the 5 binary-class SVM models. The optimal pairs of C and γ are C = 30.5 and γ = 0.00115; Therefore, the optimal parameters are used for constructing the models for future testing.

2.3. Prediction results

We use 3-fold cross validation to measure the performance. With 3-fold cross-validation approximately 1/3 of the data is left out while training and the remaining 2/3 part is used for testing. This is done cyclically 3-times, and the resulting prediction is thus a mean over 3 different testing sets. Based on our data set, the prediction performance by NN and SVM achieved the accuracy of 95.61% and 99.96%, respectively. In other words, both of them can successful using as the on-line predictor of excited emotion in game, and can easily replace the traditional method of finite state machine.

3. Simulation of Flocking Behavior by Human Emotions

3.1. Flocking

In the field of computer animation and game, simulation of crowd motions occupies a very important position. Reynolds [9] proposed a basic model of flocking behavior consisting of three simple rules: (1)

separation - avoid crowding neighbors, (2) alignment - steer towards average heading of neighbors and (3) cohesion - steer towards average position of neighbors. With these three simple rules, the flock moves in an extremely realistic way. The goal of our study is to develop a system to simulate the flocking behavior in game by emotions. The simulation system used the Microsoft XNA 3.1 to implement, this platform can develop PC game and TV game for Xbox 360. The system architecture of proposed simulation system is shown in Fig. 1.

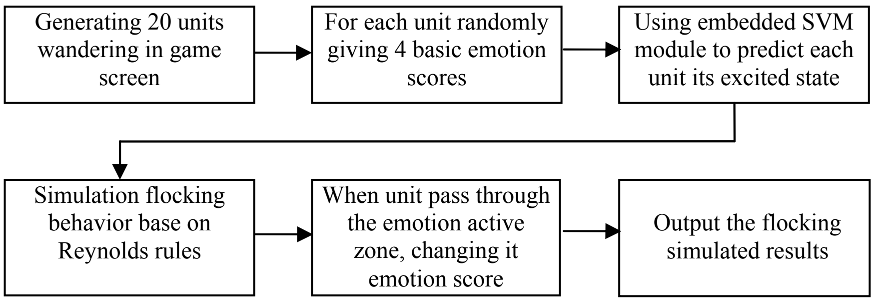

Fig. 1: System architecture of proposed simulation system

In our demonstration, we first random generate 20 units that move around (wander) in a game screen size of 800×600 pixels and each non-player character (unit) randomly gives 4 initial basic emotion scores. In addition, the normal velocity of each non-player character is set to 2 (pixel/sec), meanwhile the 4 basic emotion scores for each non-player character are updated every 2 second. Because the simulation system uses the default frame rate setting in XNA (60 frame/sec), that is, we refresh each non-player character its 4 basic emotion scores in game after 120 timeframes. According to which basic emotion is excited in game, we design a corresponding rational human behavior for non-player character to act, as shown in Table 1 to simulate the flocking.

In our simulation system, the initial generating units will first randomly wander in game screen with a normal velocity of 2 (pixel/sec). And all units their flocking behavior will obey the Reynolds rules all the time. In simulation process, the users can open the emotion active zone anytime, and move it to anywhere in the game screen. When any unit (non-player character) passes through the emotion active zone, it will change its behavior and action according to Table 1.

Table. 1: The list of NPC excited emotion vs corresponding rational behavior

Excited emotion	Design behavior in game	Design action in game	Display color
Nature	Normal (wandering in game screen)	Moving by normal velocity 2 (pixel/sec)	White
Joy	Happy (increase moving velocity)	Velocity from normal increasing to 5 (pixel/sec)	Blue
Distress	Near death (decrease moving velocity)	Velocity from normal decreasing to 1 (pixel/sec)	Brown
Anger	Fury (accelerated moving velocity)	V = Vi + 5 (pixel/sec)	Red
Fear	Running away	Changing moving direction	Green

3.2. Simulation results

The results of simulation of flocking behavior in game by human emotions are shown in Fig. 2 to Fig. 5. Fig. 2 is the simulation result for crowd (units) pass through the joy excited zone. The screenshot number 1 shown group have not yet influenced by the excited zone of joy. After that, the screenshot number 2 to 7 shown units increase their velocity and faster pass through the zone by the excited of joy, but they also keep their rank by Reynolds rules. Fig. 3 is the simulation result for crowd (units) pass through the distress excited zone. The screenshot number 2 to 5 shown units decrease their velocity and huddle together by the influence of distress. But they still keep their rank by Reynolds rules. After that, units return to the normal state of flocking.

Fig. 4 is the simulation result for crowd (units) pass through the fear excited zone. The screenshot number 2 to 6 shown units run away to avoid touch the fear excited zone. Though just a few units excited by fear, crowd immediately make a sharp turn by the flocking rule of Reynolds. After that, units return to the normal state of flocking. Fig. 5 is the simulation result for crowd (units) pass through the anger excited zone. The screenshot number 2 to 6 shown when a unit is excited by anger, it quickly rushes to pass the excited zone. Therefore, the rank formed by units is broken into pieces. After that, units return to the normal state of flocking.

4. Conclusions

The simulation of flocking behavior by human emotions in game has been few studies in these days. In this paper, we used the trained SVM module as an excited emotion predictor to embed in our simulation system. Then we simulate the flocking behavior by human emotions in game. Simulation results show that when the flock is activated by a special type of emotion, e.g. angry, our simulation system can successfully simulate the non-player characters all have the human-like (or animal-like) behavior. In other words, our emotion simulation system can embed into games to simulate non-player characters their flocking behavior.

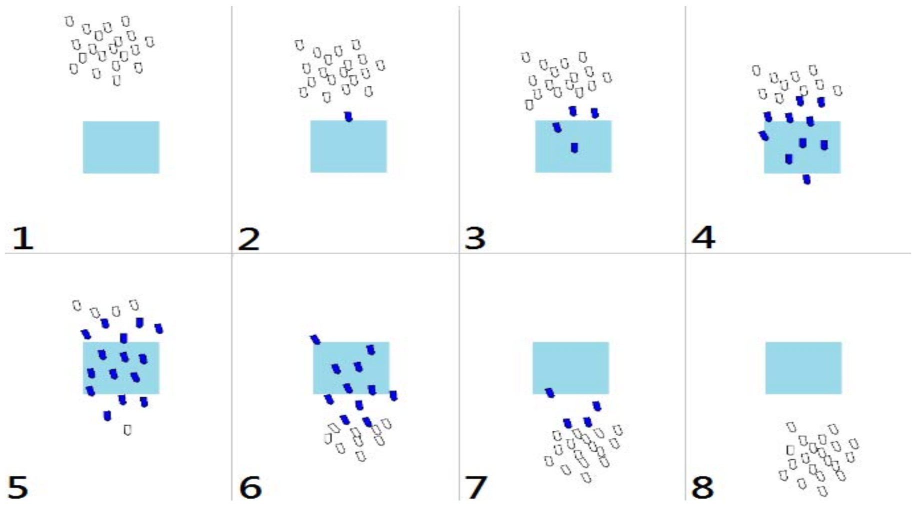

Fig. 2: Result of excited by joy

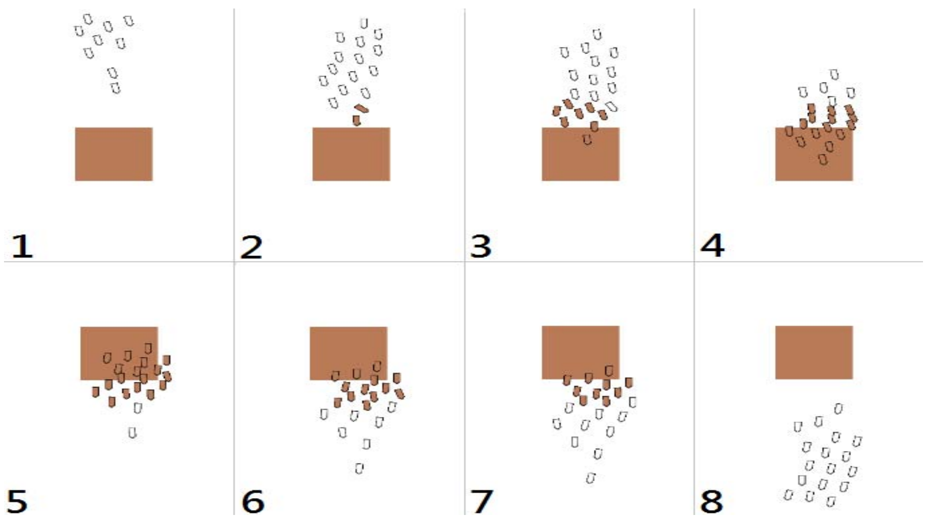

Fig. 3: Result of excited by distress

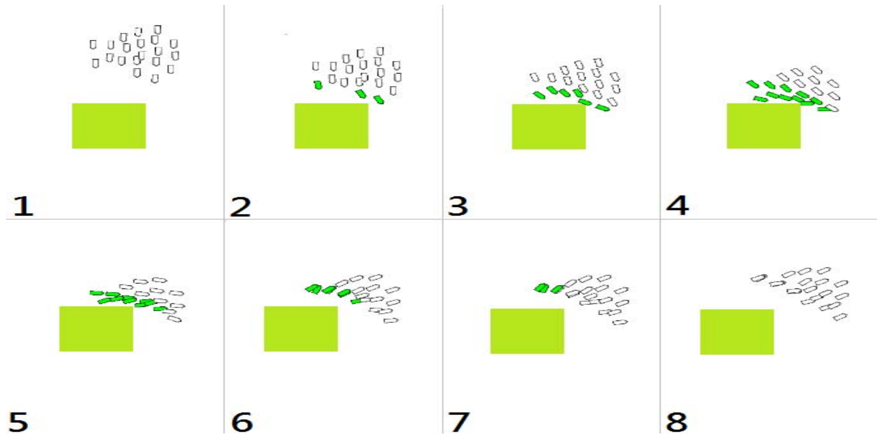

Fig. 4: Result of excited by fear

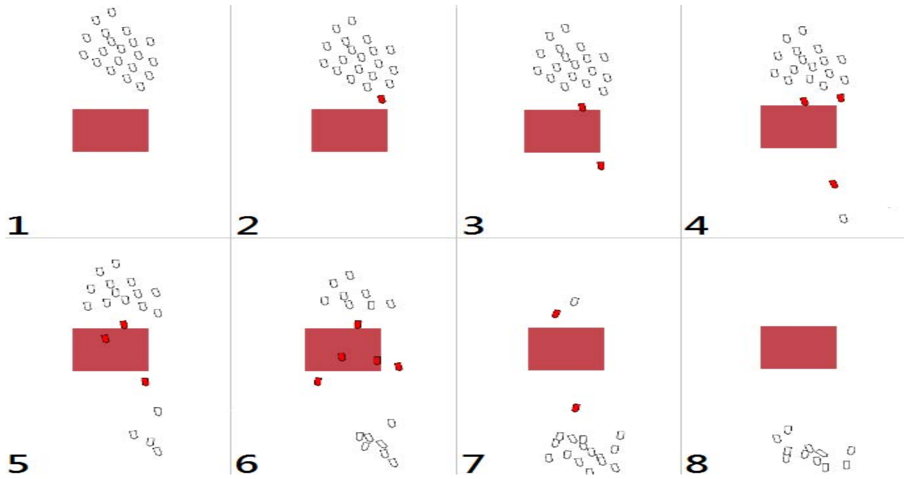

Fig. 5: Result of excited by anger

5. Acknowledgements

This research was founded in part by the National Science Council (NSC) of Taiwan under contract no. NSC 100-2221-E-262-026.

6. References

[1] D. Freeman, Creating Emotion in Games: The Craft and Art of Emotioneering, 1st ed., New Riders Games edition, 2003.

[2] M. M. Olsen, K. Harrington, and H. Siegelmann, “Emotions for Strategic Real-Time Systems,” AAAI Spring Symposium on Emotion, Personality, and Social Behavior, 2008.

[3] R. Bernhaupt, A. Boldt, T. Mirlacher, D. Wilfinger, and M. Tscheligi, “Using Emotion in Games: Emotional Flowers,” Design, pp. 41-48, 2007.

[4] M. Aqel, P. Mahanti, and S. Banerjee, “Analyzing Artificial Emotion in Game Characters Using Soft Computing,” World Academy of Science, Engineering and Technology, vol. 53, pp. 926-929, 2009.

[5] S. S. Tomkins, Approaches to Emotion, In K. R. Scherer & P. Ekman ed., Hillsdale, NJ: Erlbaum, 1984, pp. 163-195.

[6] A. Ortony, and T. J. Turner, “What's basic about basic emotions,” Psychological Review, pp. 315-331, 1997.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” SIGKDD Explorations, Vol. 11, no 1, 2009.

[8] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 1-27, 2011.

[9] C. W. Reynolds, “Flocks, Herds, and Schools: A Distributed Behavioral Model,” Computer Graphics, SIGGRAPH’87 Conference Proceedings, vol. 21, no. 4, pp. 25-34, 1987.