

## Research Article

# Computerized Interactive Gaming via Supporting Vector Machines

Y. Jiang, J. Jiang, and I. J. Palmer

*Department of Electronic Imaging and Media Communications, University of Bradford, West Yorkshire BD7 1DP, UK*

Correspondence should be addressed to J. Jiang, [j.jiang1@bradford.ac.uk](mailto:j.jiang1@bradford.ac.uk)

Received 30 June 2008; Accepted 2 November 2008

Recommended by Narendra Chaudhari

Computerized interactive gaming requires automatic processing of large volume of random data produced by players on spot, such as shooting, football kicking, and boxing. This paper describes a supporting vector machine-based artificial intelligence algorithm as one of the possible solutions to the problem of random data processing and the provision of interactive indication for further actions. In comparison with existing techniques, such as rule-based and neural networks, and so forth, our SVM-based interactive gaming algorithm has the features of (i) high-speed processing, providing instant response to the players, (ii) winner selection and control by one parameter, which can be predesigned and adjusted according to the needs of interaction and game design or specific level of difficulties, and (iii) detection of interaction points is adaptive to the input changes, and no labelled training data is required. Experiments on numerical simulation support that the proposed algorithm is robust to random noise, accurate in picking up winning data, and convenient for all interactive gaming designs.

Copyright © 2008 Y. Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

Over the past decades, artificial intelligence in computer gaming development has become an increasingly important research area, and many researchers attempted and reported algorithms across a number of disciplines including interactive gaming, character behaviour control, events and activity simulation. Theoretical and practical advancements have been made in building human-level AI systems, which enable data from other entertainment sources such as “star trek” and “star wars”, to seamlessly integrate with all level human capabilities. As a result, interactive gaming assisted with AI techniques has evolved as one of the most important research topics within the gaming industry as well as the research community [1–3]. In a stream of reports, papers, and research seminars, Laird and van Lent [1] addressed this topic on the basis of reviewing existing research on AI and gaming to describe different game genres and to specify the roles that human-level AI could play within these genres. In [4], neural network-based approaches are reported to provide intelligence for characters in fighting action games. The described approaches include three phases of designs, which are basic game rule learning and matches against

randomly acting opponents, evaluation of decision fitness by using the relative score change caused by the decision, and training of neural network by using the score difference and the previous input and output values which induced the decision. The opponents’ past actions are also utilized to find out the optimal counteractions for the patterns, and experiments show that significant understanding of the game rules is achieved by the proposed AI algorithm. AI techniques are also playing leading roles for character-based design and development [5, 6], which is essentially organized around the topic of artificial characters. The work reported in [6] introduces an intermediate architecture that fits between games and AI elements, and assesses the feasibility of improving quality of game AI.

Essentially, the AI system looks over the shoulder of players while they are taking part and learns how they deal with issues incurred in fighting games, interactive games, and all other games which require significant speed in responding and reactions. As all games are required to target a range of different players, however, capability to adapt to such changes of players and thus their reactions is essential for any AI system design. For interactive games, one of the major issues is to determine the interaction points out

of the achievement data produced by players, which can be modelled by a set of  $N$ -dimensional random points. The randomness comes from the fact that performances delivered by players are generally subject to certain levels of statistics and probabilities. Therefore, such interaction design problem can be modelled as an outlier detection, where some dataset is drawn from an underlying probability distribution, and we want to estimate a simple subset  $S$  of the input space as such that a test point being outside  $S$  can be characterized by a priori.

As one of the most popular machine learning approaches, SVM has received tremendous attention in a number of areas to deal with learning, training, and optimizing problems. General SVM uses a kernel-mapping technique to separate linearly nonseparable cases in high-dimensional space [7]. SVM not only separates data from different classes, but also separates data to its maximum margin. In other words, SVM not only divides mixed datasets into classes, but also optimizes such a classification. However, the weakness of general SVM lies in the fact that it requires labeled datasets to get it trained, yet interactive game design requires close-to-real randomness and thus responses made by players or computer-simulated opponents could be very diversified. To this end, the data produced by such random responses could be difficult to label, or manual labeling could require significant resources and time consumption. To this end, we introduce a one-class-SVM and design an AI mechanism, which can automatically process the random response datasets without any labeling process, yet capable of detecting those characteristics out of the response data randomly generated by either players or computer-simulated opponents.

One-class-SVM is essentially an extension of support vector machines [8] used for detecting the outliers [7, 9]. The idea is that it first maps the data into high-dimensional space, and then maximizes the margin between the mapped data and the origin. The main difference from general SVMs is the fact that a constraint parameter  $\nu$  is introduced to control the maximum percentage of outliers in the dataset, which can also be used to indicate the priori. As a result, the one-class-SVM is capable of being adaptive to input changes or different players when used to design the interactive gaming. This is because the priori specifies the maximum likelihood that outlier detections can make, and hence such detection is less sensitive to changes of inputs generated by different players. In comparison with neural network-based approaches, SVM presents a range of advantages, which can be summarized as: (i) while artificial neural networks (ANNs) can suffer from multiple local minima, the solution to an SVM is often global and unique; (ii) while ANNs use empirical risk minimization, SVMs use structural risk minimization. As a result, the computational complexity of SVM is not dependent on the dimensionality of the input space, and SVMs can also have a simple geometric interpretation and can generate a sparse solution; (iii) from wide range of reports on evaluation of both SVMs and ANNs [10, 11], it is generally concluded that SVMs often outperform ANNs in many classification-relevant applications due to the fact that they are less prone to overfittings.

The rest of the paper is organised into three further sections, where Section 2 is to describe the proposed AI design for interactive gaming based on one-class-SVM. Section 3 reports the experimental results, and finally, Section 4 makes concluding remarks.

## 2. BASIC CONCEPT REVIEW OF SUPPORT VECTOR MACHINE (SVM)

Supporting vector machine is a powerful classification tool widely used in many areas of research and applications, especially for pattern recognition and classifications [7, 9, 12]. Its basic concept can be best illustrated in a linear classification with only two classes:  $\omega_1$  and  $\omega_2$ . Given  $N$  training vectors,  $\{x_1, x_2, \dots, x_N\}$ , being fed into an SVM, each vector will have an output indicator  $y_i = \{y_1, y_2, \dots, y_N\}$ , and a hyperplane  $g(x) = w^T x + w_0$  needs to be determined, which can achieve the following classifications:

$$\begin{aligned} \text{for } x_i \in \omega_1 : g(x_i) = w^T x_i + w_0 &\geq 1 \quad (y_i = 1), \\ \text{for } x_i \in \omega_2 : g(x_i) = w^T x_i + w_0 &\leq -1 \quad (y_i = -1), \end{aligned} \quad (1)$$

where  $w = (w_1, w_2, \dots, w_N)$  is referred to as supporting vectors as they comprise the basic elements of the hyperplane for such binary classification. To optimize the classification,  $w$  and  $w_0$  need to be determined as such that the cost function  $J(w) \equiv (1/2)\|w\|^2$  is minimized, subject to the condition:  $y_i(w^T x_i + w_0) \geq 1$ .

To solve the above problem, the technique of Lagrange multipliers is generally used, where a Lagrange function is introduced via combination of the cost function with its constraint condition. This is defined as

$$L(w, w_0, \lambda) = \frac{1}{2} w^T w - \sum_{i=1}^N \lambda_i [y_i (w^T x_i + w_0) - 1]. \quad (2)$$

Therefore, minimizing the cost function is equivalent to maximizing the Lagrange function, and its corresponding solutions satisfy the following conditions [7, 9]:

$$\frac{\partial L(w, w_0, \lambda)}{\partial w} = 0, \quad (3)$$

$$\frac{\partial L(w, w_0, \lambda)}{\partial w_0} = 0, \quad (4)$$

$$\lambda_i [y_i (w^T x_i + w_0) - 1] = 0, \quad (5)$$

$$\lambda_i \geq 0. \quad (6)$$

From the above conditions, the supporting vectors can be worked out as

$$w = \sum_{i=1}^N \lambda_i y_i x_i. \quad (7)$$

The Lagrange multipliers  $\lambda_i$  can be determined from

$$\sum_{i=1}^N \lambda_i y_i = 0. \quad (8)$$

To illustrate the essential concept of SVM, we use a simple linear example, in which a training set consists of four points with two classes given below

$$\begin{aligned} \omega_1 &= \{x_1 = [1, 1]^T, x_2 = [1, -1]^T\}, \\ \omega_2 &= \{x_3 = [-1, 1]^T, x_4 = [-1, -1]^T\}. \end{aligned} \quad (9)$$

With the Lagrange multiplier approach as explained above, the following equations can be established from (7) and (8) since in this case we have  $y_1 = y_2 = 1$  and  $y_3 = y_4 = -1$  as follows:

$$\begin{aligned} w_1 &= \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4, \\ w_2 &= \lambda_1 + \lambda_4 - \lambda_2 - \lambda_3 \\ \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 &= 0. \end{aligned} \quad (10)$$

According to (5), however, we have

$$\begin{aligned} \lambda_1(w_1 + w_2 + w_0 - 1) &= 0, \\ \lambda_2(w_1 - w_2 + w_0 - 1) &= 0, \\ \lambda_3(w_1 - w_2 - w_0 - 1) &= 0, \\ \lambda_4(w_1 + w_2 - w_0 - 1) &= 0. \end{aligned} \quad (11)$$

As (11) gives us  $w_1 + w_2 = 1$  and  $w_1 - w_2 = 1$ , solutions to these two simultaneous equations give us  $w_1 = 1$  and  $w_2 = 0$ . By putting these solutions back to (11), it can be derived that  $w_0 = 0$ . As a result, the optimized hyperplane for dividing these two classes can be determined as  $g(x) = w^T x + w_0 = w_1 x_1 + w_2 x_2 + w_0 = x_1 = 0$ . Indeed, observations can easily reveal that all the four points become the supporting vectors, and the margin of the separating hyperplane (line) from both classes is equal to 1.

For nonlinear SVM-based classification, the concept is to map the input vectors  $x \in R^l$  into another  $k$ -dimensional feature space:  $y \in R^k$  (normally  $k > l$ ). Consequently, the SVM classification becomes

$$\begin{aligned} \text{Assign } x \text{ in } \omega_1 \text{ if } g(x) &= \sum_{i=0}^{N_s} \lambda_i y_i K(x_i, x) + w_0 > 0, \\ \text{Assign } x \text{ in } \omega_2 \text{ if } g(x) &= \sum_{i=0}^{N_s} \lambda_i y_i K(x_i, x) + w_0 < 0, \end{aligned} \quad (12)$$

where  $N_s$  is the total number of supporting vectors,  $K(x_i, x)$  is known as a kernel function. When the input  $x \in R^l$  is mapped into another feature space via  $\phi: x \rightarrow \phi(x) \in \psi$ , the inner product operation has an equivalent representation  $\sum_r \phi_r(x) \phi_r(z) = K(x, z)$ , where  $\phi_r(x)$  is the  $r$ -component of the mapping  $\phi(x)$  of  $x$ . Typical examples of the kernel functions used in SVM include (i) polynomials:  $K(x, z) = (x^T z + 1)^q$ ,  $q > 0$ ; (ii) radial basis functions (RBF):  $K(x, z) = \exp(-\|x - z\|^2 / \sigma^2)$ , which is also referred to as Gaussian kernel; and (iii) hyperbolic tangent:  $K(x, z) = \tanh(\beta x^T z + \gamma)$ . Formulation of SVM does not include criteria to select a kernel function that gives good generalization, detection,

and classification. Its selection is dependent on the dataset to be classified and applications involved. The general principle is to use Gaussian as the RBF and its variance becomes essential in SVM design, which can often be estimated and assessed via training data. On the other hand, the polynomial kernel function is directional, that is, the output depends on the direction of the two vectors in low-dimensional space. Extensive work has been carried out and reported regarding the kernel selection, which can be generally classified as (i) entropy-based approach [13] and (ii) empirical approach [11], where part of training data is used to assess the effectiveness of the kernel candidates. In our proposed algorithm, we used Gaussian as the RBF kernel for our SVM design.

As seen, the principle for linear SVM and nonlinear SVM remains the same, and the only difference is that a feature space mapping  $x \rightarrow \phi(x)$  is applied to transform the nonlinear case into a linear one, and thus the hyperplane can be determined through the inner product of these mapping functions, that is, the kernels, since the kernels are readily selected as one of the above three.

### 3. THE PROPOSED SVM INTERACTION DESIGN

The scenario for interactive gaming is that among all the responses generated by players or the simulated computer opponents, only a small proportion represents high level hits, which need to be detected and picked up for the design of reactions. This is a basic mechanism to make the game playing more challenging, which can attract players to learn and improve their skills. Such a scenario can be described by a dataset with  $T = \{x_1, x_2, \dots, x_l\}$ ,  $x \in R^N$ , as shown in Figure 1 for a two-dimensional case ( $N = 2$ ). The task is to find a function  $f$  that generates the value “+1” for most of the vectors in the dataset (marked as stars in Figure 1(a)), and “-1” for the other very small part (marked as circles in Figure 1(a)). The strategy for such a classification and detection is to use a one-class-SVM [8] and to map the input data into a Hilbert space  $H$  according to a mapping function  $X = \phi(x)$ , as shown in Figure 1(b), and to separate the data from the origin to its maximum margin.

As a result, to separate the mapped data from the origin to its maximum margin is equivalent to solving the following quadratic optimization problem:

$$\min_{w \in F} \frac{1}{2} \|w\|^2 + \frac{1}{\nu l} \sum_i \xi_i - \rho. \quad (13)$$

Subject to

$$f(x) = w \phi(x_i) - \rho \geq -\xi_i, \quad \xi_i > 0, \quad i = 1, \dots, l, \quad (14)$$

where  $\nu \in (0, 1)$  is a constraint parameter to limit the maximum proportion of the high performance responses among all the ordinary responses such that a maximum of  $\nu \times 100\%$  is expected to return negative values according to  $f(x) = w \cdot \phi(x) - \rho$ .  $\xi_i$  are slack variables acting as penalties in the objective function. Therefore, based on its dual representations, the one-class-SVM is to solve the following problems.

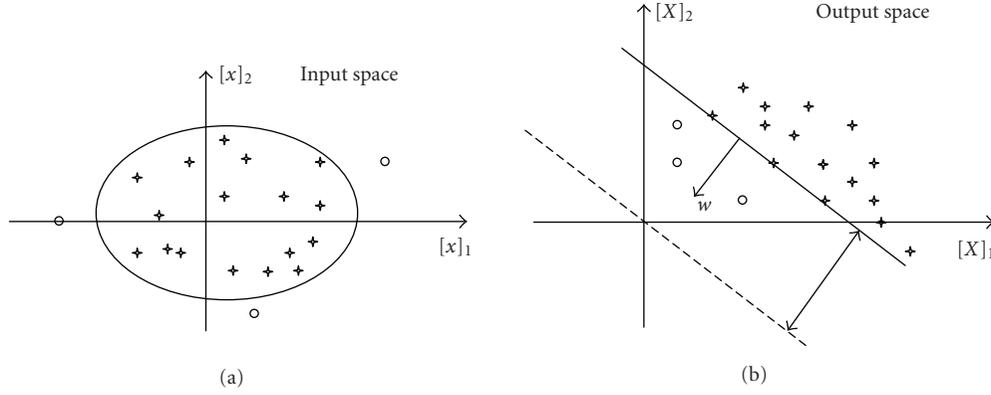


FIGURE 1: Data mapping by (a) one-class-SVM from input space to (b) output space.

- (1) Select a kernel function  $K(x, x')$  in Hilbert space  $H$ , and the sensitivity parameter  $\nu$  to solve the following optimization problem and find the solution of  $\alpha^* = (\alpha_1^*, \dots, \alpha_l^*)$  as follows:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j K(x_i, x_j). \quad (15)$$

Subject to

$$0 \leq \alpha_i \leq \frac{1}{(\nu \cdot l)}, \quad i = 1, \dots, l, \quad (16)$$

where  $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$  is the kernel function, and  $\sum_{i=1}^l \alpha_i = 1$ .

- (2) Select any  $\alpha^*$  which satisfies  $0 < \alpha^* < 1/(\nu \cdot l)$ , and calculate the bias  $\rho = \sum_{i=1}^l \alpha_i^* K(x_i, x_j)$ . The vectors that satisfy  $0 < \alpha^* < 1/(\nu \cdot l)$  are called support vectors.
- (3) Integrate decision function  $f(x) = \sum_{i=1}^{N_{sv}} \alpha_i^* K(x_i, x) - \rho$ , and if  $f(x) \geq 0$ , it returns +1. Otherwise, return the real negative value, where  $N_{sv}$  is the number of support vectors.

It is proved in [8] that  $\nu \times 100$  is the upper bound percentage of the data that are expected to be outliers in the training data, and a vector  $x_i$  is detected to be outlier in the training set, if and only if  $\alpha_i = 1/(\nu \cdot l)$ .  $\alpha_i$  is the parameter that directly determines the sensitivity of outlier detection using one-class-SVM. Its selection is dependent on the specific interactive game design and specific setting for the level of difficulties, which is often dependent on individual players. In our simulated experiments given in the next section, we constrained the maximum number of interactive points to be detected at 20%.

Figure 2 illustrates the structure of our proposed AI interactive game design, where the one-class-SVM is taken as the core of the response data processing. Depending on the response data capture mechanisms, further data preprocessing can be designed to improve the SVM-based response data selection and classification. In our simulated

experiments, we designed the input response vector to contain the location of the response, and the strength of the response. As a result, the input data is a vector of three elements. Following the SVM-based data detection, a basic clustering [14] is designed as a postprocessing to cluster the detected response data into three categories, outstanding, excellent, and good, to enable the game designers to allocate appropriate awards to the players' performances. Essentially, all the three categories also represent the interactive points, upon which further interactive actions can be designed accordingly.

#### 4. SIMULATED EXPERIMENTS AND EVALUATION OF THE PROPOSED AI DESIGN

In this section, numerical experiments are carried out to demonstrate the effectiveness of the SVM-based response data processing and the interactive gaming design. To evaluate the proposed AI interactive algorithm, we designed a simple interactive game to collect the training and testing data, where response is simulated by a pseudorandom generation on a computer screen. While its hit-on-screen location is represented by two coordinates  $(x, y)$ , its hitting strength can be represented by the intensity value of its display on a grey level scales (0 ~ 255). To simulate the different contexts and the varying nature of the input data content, we randomly added bias data into the input data capture and asked different players to generate the random input vectors. Following that, performance evaluation and interaction are simulated by the proposed SVM-based system as illustrated in Figure 2.

Figure 3 illustrates the simulated experimental results, where the pink colour highlighted points are the high-performing responses automatically detected by our proposed algorithm as the interactive points that need to be further classified to determine the level of rewards for the corresponding players, and those blue colour highlighted points are ordinary responses that do not receive any rewards. As a matter of fact, those high-performing responses represent the set of interactive points for general game design, where interactive actions can be further

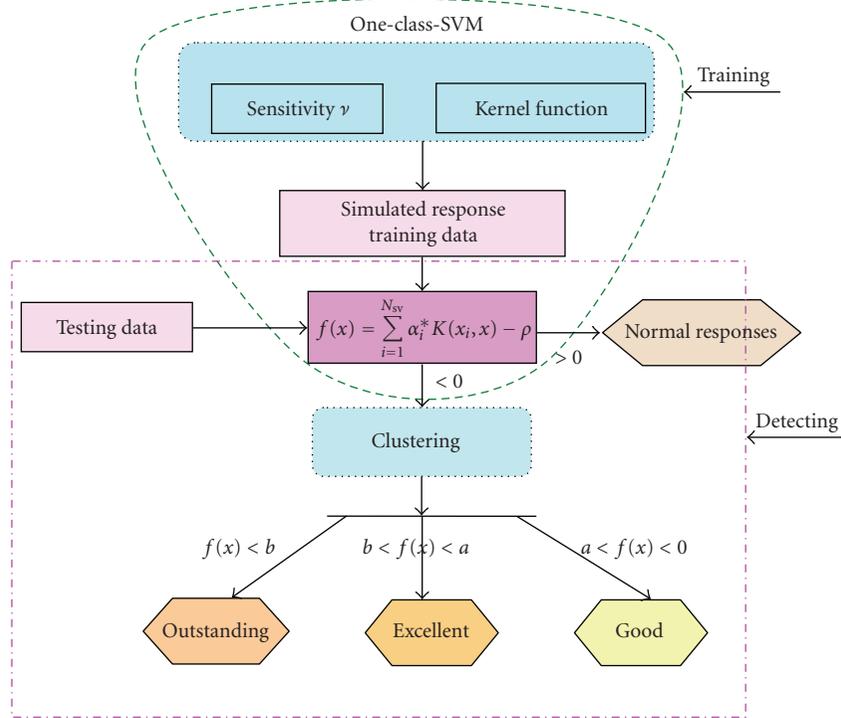


FIGURE 2: Structural illustration of the SVM-based AI interaction design.

designed according to the specific needs of the game. The experimental results illustrated in Figure 3 prove that the proposed algorithm is capable of detecting those interactive points or high performance responses automatically without any regards to who the players are. As seen, not only those high-performing responses are effectively detected by the SVM-based AI algorithm, but also their context is adapted to automatically by the proposed system.

Specifically, part (a) in Figure 3 represents a less skilled player, whose high performance responses are detected by the proposed AI algorithm with an equivalent threshold at around 38. The player who generated the experimental results in part (b) of Figure 3 is better skilled, where his high performance responses are detected with an equivalent threshold at around 70. Further observation reveals that the part (c) and part (d) of Figure 3 represent even better skilled players, where their high performances are detected with equivalent thresholds at around 81 and 85, respectively. With the proposed algorithm, no labelled data is required for both training and testing. The proposed algorithm is capable of automatically adapting to the different players and detecting their high performance responses. In contrast, existing rule-based approaches [12] would have to set up different thresholds for all the four players. With general SVM or neural network approaches [4, 13], separate training with labelled data for each of the four players will also be required in order to make them adaptive to other unseen players with different skills. As interaction is desired to be tailored for individual players, the advantage that the proposed AI algorithm is not only capable of detecting high performance responses (outliers) but also adapting to

their context changes would prove very useful for practical interactive game designs and developments.

In addition, the proposed algorithm is fully automatic and computerized, in which only a few initial sample testing is needed depending on specific gaming design, and such initial sample testing is often helpful for the designed interactive system to collect training data and thus to get the SVM element trained to allow the system to learn from the players, especially when the player is first time encountered.

## 5. CONCLUSIONS

While existing AI techniques developed for interactive gaming are mostly focused on neural networks and rule-based approaches, we propose an SVM-based approach in this paper to illustrate the effectiveness and efficiency in processing responses and to generate reactions. Compared with other AI techniques, the proposed has the features of: (i) self-adaptive to the response data content by controlling one trade off parameter; (ii) detection of high-performing responses according to the context the responses are generated; (iii) training does not require any labeled data, which can be very useful for circumstances where manual labeling is labour intensive and time consuming. Following the proposed AI design, a number of options can be further designed for specific reactions to the detected responses, such as awards, penalties, or any other actions that can further stimulate responses. Therefore, the proposed AI design can have many useful applications in practical computer game design when interaction is required.

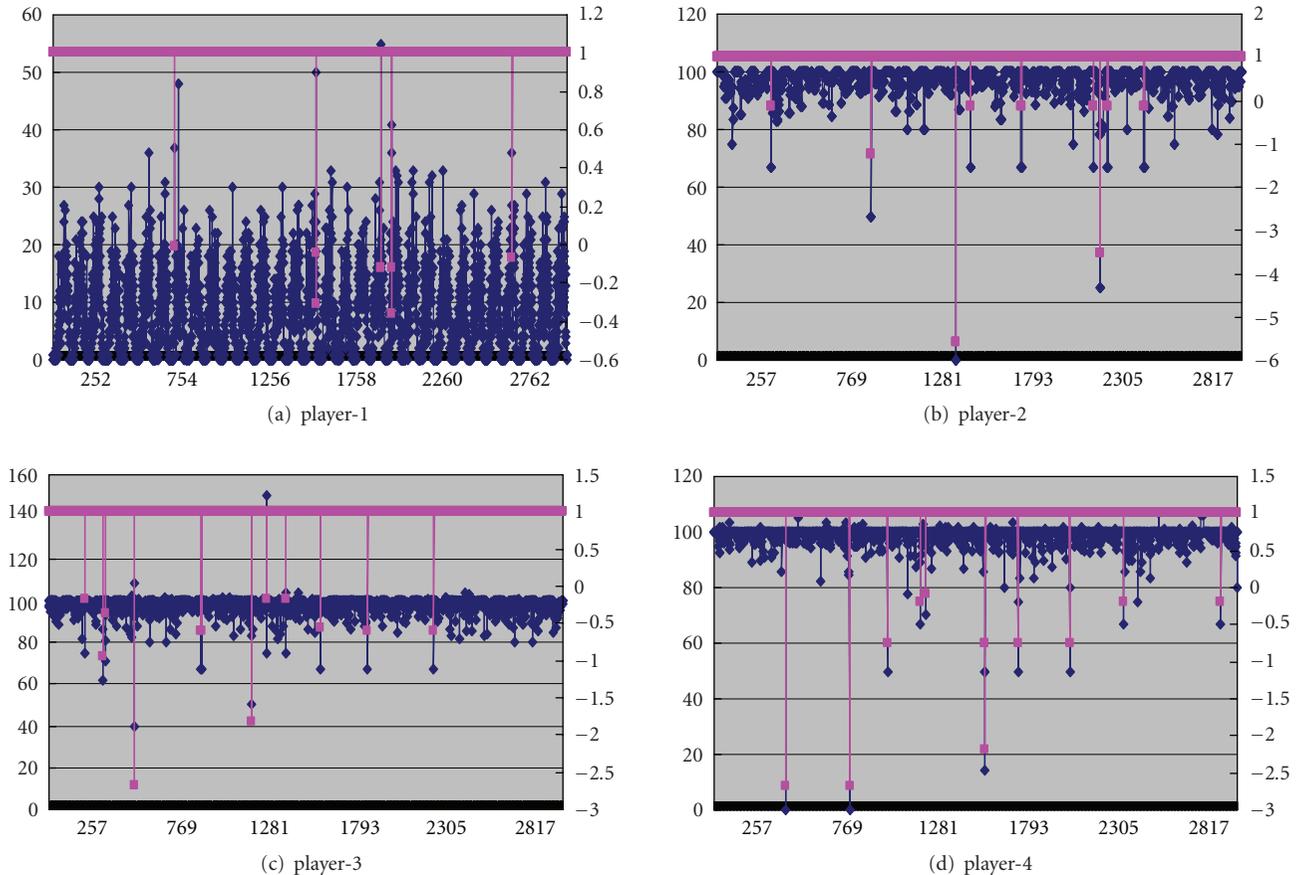


FIGURE 3: Summary illustration of experimental results.

## ACKNOWLEDGMENT

The authors wish to acknowledge the financial support under the HERMES project funded by European FP7 Programme with Contract IST-216709.

## REFERENCES

- [1] J. E. Laird and M. van Lent, "Interactive computer games: human-level AI's killer application," in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI '00)*, Austin, Tex, USA, July-August 2000.
- [2] J. E. Laird, "Using a computer game to develop advanced AI," *Computer*, vol. 34, no. 7, pp. 70–75, 2001.
- [3] B. H. Cho, S. H. Jung, Y. R. Seong, and H. R. Oh, "Exploiting intelligence in fighting action games using neural networks," *IEICE Transactions on Information and Systems*, vol. E89-D, no. 3, pp. 1249–1256, 2006.
- [4] B. Magerko, J. E. Laird, M. Assanie, A. Kerfoot, and D. Stokes, "AI characters and directors for interactive computer games," in *Proceedings of the 16th Innovative Applications of Artificial Intelligence Conference (IAAI '04)*, pp. 877–883, AAAI Press, San Jose, Calif, USA, July 2004.
- [5] M. Dawes and R. Hall, "Towards using first-person shooter computer games as an artificial intelligence testbed," in *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES '05)*, vol. 3681 of *Lecture Notes in Computer Science*, pp. 276–282, Melbourne, Australia, September 2005.
- [6] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 1741–1745, Portland, Ore, USA, July 2003.
- [7] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [8] B. Schölkopf, R. Williamson, A. J. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*, pp. 582–588, MIT Press, Cambridge, Mass, USA, 2000.
- [9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 2000.
- [10] I. Kalatzis, N. Piliouras, E. Ventouras, C. C. Papageorgiou, A. D. Rabavilas, and D. Cavouras, "Comparative evaluation of probabilistic neural network versus support vector machines classifiers in discriminating ERP signals of depressive patients from healthy controls," in *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis (ISPA '03)*, vol. 2, pp. 981–985, Rome, Italy, September 2003.
- [11] Y. Li and J. Jiang, "Combination of SVM knowledge for microcalcification detection in digital mammograms," in *Proceedings of the 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL '04)*, vol.

- 3177 of *Lecture Notes in Computer Science*, pp. 359–365, Springer, Exeter, UK, August 2004.
- [12] Y. Weng and J. Jiang, “Real-time and automatic close-up retrieval from compressed videos,” *International Journal of Automation and Computing*, vol. 5, no. 2, pp. 198–201, 2008.
- [13] M. Barrientos, H. de Bruin, and B. L. Bardakjian, “Criteria to select kernel functions for positive time-frequency distributions of theoretical time-varying signals,” in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 209–212, Philadelphia, Pa, USA, October 1994.
- [14] J. Jiang, “A parallel computing and neural network implementation of LBG imagevector quantization,” in *Proceedings of the 6th International Conference on Image Processing and Its Applications (IPA '97)*, vol. 1, pp. 27–31, Dublin, Ireland, July 1997.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

