Mimicking player strategies in fighting games

Simardeep S. Saini, Christian W. Dawson, and Paul W. H. Chung
Department of Computer Science,
Loughborough University,
Loughborough, UK, LE11 3TU
S.S.Saini@lboro.ac.uk

Abstract-- **The paper presented here provides an account of the research carried out in the field of Game Artificial Intelligence (AI) related to mimicking human player strategies in fighting games. The problem is introduced followed by a description of the Proof of Concept game that is used as a test bed for the implementation of a novel AI system. An overview of the system is also provided with the detailed results of a demonstration that alludes to the effectiveness of the system.**

I. INTRODUCTION

The increasing popularity of online multiplayer games has given rise to a new forum for the use of Artificial Intelligence (AI) to mimic human players. Many fighting games are now available for playing online against other human players, allowing for fresh play styles and challenges. Fighting games can typically be played as either a single player experience, or against another human player, whether it is via a network or a traditional multiplayer experience. However, there are two issues with these approaches. First, the single player offering in many fighting games is regarded as being simplistic in design, often relying upon Finite State Machines, making the moves predictable [1]. Secondly, while new challenges and unique fighting strategies can be provided by having human players play one another, this may not always be achievable due to the logistics involved in setting up such a bout and the availability of both players. *Game AI* could provide a solution to both of these issues, allowing a human player's strategy to be learned and then mimicked by the CPU fighter.

While much research within the field of Game AI focuses on improving the AI and providing more of a challenge to the player [2], research on mimicking human player strategies is limited. The research conducted here focuses on the latter.

In related research, machine learning was used to classify tactics to pre-defined states in a data driven finite state machine which was referred to in real time to mimic the overall strategy of a human player [3]. However, the pre-defined states made for a rigid framework that could only cater for a small subset of strategies. This restriction drastically limits the usage of the aforementioned architecture. Furthermore, only the health parameter is used to trigger state transitions, as opposed to mult-parameters used in many modern fighting games.

The novel architecture proposed in this paper addresses these issues by combining data driven finite state machines, unsupervised learning and classification within a multi-parameter fighting game. Strategies and the corresponding tactical moves are mimicked such that the CPU player is able to play against another human player by utilizing the strategy learned from another human opponent. A proof of concept fighting game based on this approach has been designed and created as a test bed that lends itself to strategic gameplay.

II. PROOF OF CONCEPT

The proof of concept game is a basic one-on-one, hand-to-hand fighting game. Players' moves are restricted to a multitude of attack techniques that vary in range, damage and speed. A growing trend in fighting games since the mid 1990s is the use of multiple parameters to add a strategic element to the gameplay. Capcom's *Street Fighter Alpha 3* makes use of three separate parameters per fighter; health, the block gauge (which caps the amount of blocking done consecutively), and the super combo meter (which builds up throughout gameplay, allowing for powerful attacks to be performed). The proof of concept game is in the same vein with three parameters in use per player; health, morale and stamina. This makes for a total of six game parameters. Health depletes as players endure damage; stamina depletes as players perform moves, blocks and attacks; and morale is increased with successful evasions. As morale increases beyond certain thresholds, so does the proportion of damage dealt to the opponent. Each player's health and stamina is initiated at 100, while morale is initiated at 50.

Further to attacking, players can perform low or high blocks and a variety of evasions. Certain attacks are blocked by using the high block, while others can only be blocked using the low block. The various moves and their effects within the game are presented in Table I.

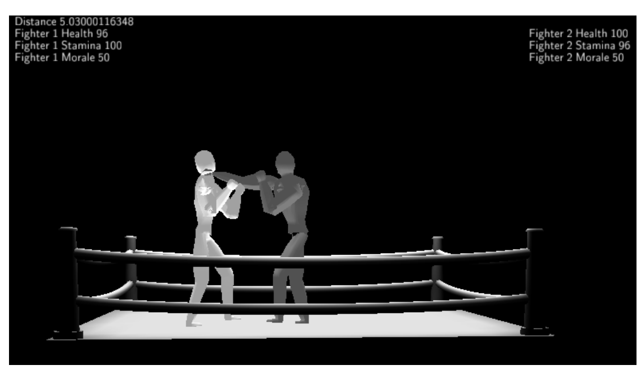

Figure 1 – Screenshot of Proof of Concept game.

TABLE I
GAME MOVES

Move	Distance From	Distance To	Health Depleted	Stamina Depleted	Morale Gained for Evasion	Blocked	Evasion	Notes
Jab	4.1	5	1	1		Stam - 1	Back	
Cross	4.1	5.5	2	2		Stam – 1	Left	
Right Hook	4	4.7	3	2		Stam – 1	Back	
Left Hook	4	4.7	3	2		Stam – 1	Back	
Uppercut	0	4	4	2		Stam – 1	Right	
Haymaker	4	4.5	10	5				Unblockable
Right Body Shot	0	4	2	1		Stam – 1		
Left Body Shot	0	4	2	1		Stam – 1		
Short Jab	0	4	2	1		Stam – 1	Back	
Short Cross	0	4	3	2		Stam – 2	Left	
Evade Back					2			Evasion
Evade Left					2			Evasion
Evade Right					2			Evasion
Push	0	4	2	1		Stam – 1		Pushes opponent 5 back
Block								Blocks high attacks
Low Block								Blocks low attacks
Low Kick	0	4	2	1		Stam – 1		
Sidekick	4.1	5.5	4	2		Stam – 2		
F Lunge				5				Moves player 6 Forward
B Lunge				5				Moves player 6 Back

TABLE II
VECTOR CALCULATION

Move	x0 (No. of Moves)	x1 (Total Damage)	x2 (Damage Ratio)	x3 (No. of Blocks)	x4 (No. of Evasions)	x5 (No. of Front Lunges)	x6 (No. of Back Lunges)	x7 (Distance between players)
Jab	+1	+1	x1 / x0					Execution distance
Cross	+1	+2	x1 / x0					Execution distance
Right Hook	+1	+3	x1 / x0					Execution distance
Left Hook	+1	+3	x1 / x0					Execution distance
Uppercut	+1	+4	x1 / x0					Execution distance
Haymaker	+1	+10	x1 / x0					Execution distance
Right Body Shot	+1	+2	x1 / x0					Execution distance
Left Body Shot	+1	+2	x1 / x0					Execution distance
Short Jab	+1	+2	x1 / x0					Execution distance
Short Cross	+1	+3	x1 / x0					Execution distance
Evade Back	+1		x1 / x0		+1			Execution distance
Evade Left	+1		x1 / x0		+1			Execution distance
Evade Right	+1		x1 / x0		+1			Execution distance
Push	+1	+2	x1 / x0					Execution distance
Block	+1		x1 / x0	+1				Execution distance
Low Block	+1		x1 / x0	+!				Execution distance
Low Kick	+1	+2	x1 / x0					Execution distance
Sidekick	+1	+4	x1 / x0					Execution distance
F Lunge	+1		x1 / x0			+1		Execution distance
B Lunge	+1		x1 / x0				+1	Execution distance

TABLE III
LIKE STATE TRANSITIONS

Prev	Curr	Next	P1H	P1S	P1M	P2H	P2S	P2M
S0	S1	S2	63	31	51	88	47	86
S0	S1	S2	61	30	40	15	34	65
S0	S1	S2	62	29	23	78	54	73
Variance			0.7	0.7	132.7	1044.3	68.7	75.0

III. SYSTEM DESIGN

The AI design uses a variety of existing techniques to mimic the human player. The game is initially played by two human players fighting one another, one of whom is to be mimicked. This is repeated numerous times, during which, data including each of the six parameters of the game, as well as the moves that are carried out, are spooled to a text file. There is an underlying assumption that the player being mimicked uses the same strategy each time.

The design addresses each of the layers of decision making with a different technique; an all-encompassing data driven finite state machine is used at the strategic level, hierarchical clustering is used at the tactical level, and Nearest Neighbour classification is used at the operational level. Fig 2 shows the flow of data as well as the stages at which various techniques are used within the system architecture.

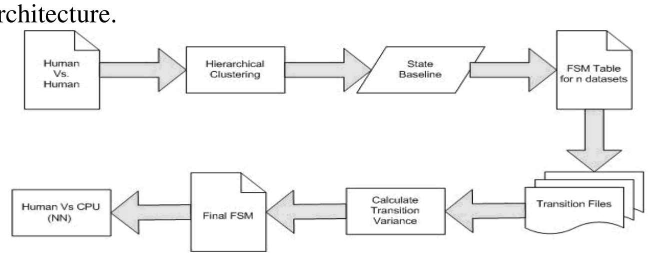

Figure 2 – AI data flow

Data are collated during the initial bouts where the two human players play against one another. These data contain various statistics based on the game parameters and moves performed. Before the moves within the collated data can be used to form the states for data driven finite state machine (DDFSM), they must be assigned some meaningful values such that like-instances can be grouped. In order to achieve this, each of the moves and combinations of moves performed are quantified to a vector X, such that X = (x0, x1, x2, x3, x4, x5, x6, x7), where x0…x7 represent the parameters listed in Table II. These vectors are then clustered using complete linkage hierarchical clustering :

$$D(X_i, X_j) = \max_{x \in X_i} \max_{y \in X_j} d(x, y) \tag{1}$$

This is where the distance between two clusters is defined as being the distance between the two furthest elements [4]. For the purposes of the research conducted here, a distance criterion of 2 has been set, beyond which clusters are not merged. The clustered datasets, s0, s1, s2,….,sn, act as states for a DDFSM, with moves and combinations of moves residing within each state.

Having established the states, the raw data from the human vs. human bouts are re-analysed and state transitions determined. As this is a multi-parameter game, the game must be played several times between the same humans using the same strategies. Upon re-analysing the data, similar state transitions are identified. This is where the previous, current and next states for one bout are the same as those for subsequent bouts. For example, all transitions across the multiple bouts where the previous state was s0, the current state is s1 and the next state is s2, would be collated. The values of each of the six game parameters for each of the similar transitions are assessed, and the variances between the parameter values in one transition and those of similar transitions are calculated. If the variance between two of the same parameters is below a threshold of 10, the parameter and its mean value is considered a transition function for that particular state transition. For example, the data shown in Table III could be considered.

Table III shows three similar transitions and the values of each of the parameters when the transition occurred during the human vs. human bouts. The variance is calculated for each of the six game parameters. Player 1 health and player 2 stamina have a variance below the threshold, therefore, it is assumed that these parameters trigger the state transition. The mean value across the three bouts for these parameters is calculated and is used as the threshold for this particular transition function. It is deduced that when player 1 health falls below 61, and player 1 stamina falls below 30, the CPU can move from state s1 to state s2, provided the previous state was s0.

Having previously clustered the moves to states, once the variances have been calculated, the DDFSM can be generated. During gameplay against the CPU fighter, once a state within the DDFSM has been entered, an operation is selected. This is achieved by calculating the Euclidean distance between the query vector ***r***, which represents real time parameters of the game, and each vector in the set ***V,*** which represents the set of vectors containing game parameters collated during the human vs. human bout.

Each state has a corresponding file containing the moves that are to be performed as well as the values of the parameters under which they had been performed during the human vs. human bouts. The values of the parameters represent the vectors belonging to ***V***, whereas the corresponding moves for each vector form the set of outputs, ***O***. The Euclidian distance between ***r*** and every element within ***V*** is calculated using the following equation:

$$d(\mathbf{r},\mathbf{v}) = \sqrt{\sum_{i=1}^{n}(r_i - v_i)^2} \tag{2}$$

The vector in ***V*** with the shortest distance to ***r*** is determined and the corresponding output from ***O*** is performed. The calculation of the Euclidean distance and selection of the output is performed during gameplay.

IV. IMPLEMENTATION

To demonstrate the effectiveness of this approach, a strategy has been formulated and played out three times in human vs. human bouts. The strategy and it's associated tactics are highlighted in Table IV.

After collating the data from the three human vs. human bouts, the clustering is performed using the complete linkage hierarchical clustering capability found in MultiDendrograms [4]. The clustering gives rise to states, each containing tactics as outlined in Table V.

Once the states have been established, like state transitions are identified and variances between the parameters amongst the like-counterparts are calculated (as described above). The DDFSM shown in Table VI is created and used during the human vs. CPU bout.

TABLE IV
STRATEGY FOR HUMAN VS HUMAN BOUT

Description	Moves Performed
Begin by performing long range moves/combinations at a distance.	[Jab, Cross] [Jab, Jab, Cross] [Jab] [Cross]
If health depletes below 68, block opponents attacks	[Block] [Low Block]
If stamina depletes below 55, begin evading the opponent's attacks.	[Evade Back] [Evade Left]
If player's morale exceeds 75, begin performing close range attacks.	[Uppercut, Right Body] [Uppercut] [Right Body, Left Body] [Low Kick, Left Body]

TABLE V
GENERATED STATES

State	Tactics
s0	[Jab, Cross] [Jab, Jab, Cross] [Jab] [Cross] [Right Body, Left Body] [Uppercut]
s1	[Block] [Low Block]
s2	[Evade Back] [Evade Left]
s3	[Uppercut, Right Body] [Uppercut] [Right Body, Left Body] [Low Kick, Left Body]
s4	[Uppercut]

TABLE VI
DATA DRIVEN FINITE STATE MACHINE

Previous	Current	Next	Transition Function
null	s0	s1	CPU Health < 67
s0	s1	s2	CPU Stamina < 52
s1	s2	s3	CPU Morale > 76

TABLE VII
REALTIME DATA SNAPSHOTS

P1 Health	P1 Morale	P1 Stam	CPU Health	CPU Morale	CPU Stam	Moves
100	50	100	100	50	99	Jab, Jab
90	50	78	81	50	89	Cross
89	50	78	81	50	88	Cross
88	50	76	78	50	87	Jab, Cross
87	50	65	66	50	84	Block
87	50	59	66	50	78	Block
87	50	52	66	50	73	Block
87	50	46	66	50	68	L Block
87	50	37	66	50	59	Block
87	50	31	66	52	53	Back
87	50	23	66	68	53	Back
87	50	20	66	74	53	Back
83	50	19	66	76	50	Upper
75	50	19	66	76	49	Upper
55	50	19	66	76	44	Upper
51	50	19	66	76	42	R.Body L.Body
31	50	19	66	76	38	R.Body
23	50	19	66	76	36	L.Body

The FSM shown in Table VI is in accordance to the strategy outlined in Table IV. When the FSM is actioned during gameplay, once within a state, the appropriate moves are selected. The Table VII contains snapshots of data at certain intervals, outlining the moves that were performed under various circumstances. Table VII shows that the moves selected at the operational level from the pool of moves within each state fall in line with the strategy outlined in Table IV, therefore demonstrating the usefulness of this technique.

V. CONCLUSION

The results of the demonstration presented in Table V, Table VI and Table VII indicate that both the tactics and the overall strategy have been successfully mimicked. There are no restrictions on the number of states that can be implemented. Furthermore, this proposed architecture can cater for multi-parameter transitions. However, there is currently no noise reduction implemented. Anomalies in the data caused by human error during the human vs. human bouts have the potential to prevent the successful application of this approach.

If a human player does not play out there strategy exactly in a number of bouts, the variance between like transitions may exceed the threshold, thus invalidating the DDFSM. Further to this, the vector calculation presented in Table II treats each value of the input vector with equal importance. There is currently no weighting, signifying which factors are more influential than others. For example, Table V shows various attacks belonging to s0, including left/right body shots and uppercut, largely because these moves were performed at the same distance as the intended long range moves. These moves were not executed during they human vs. CPU bout as the Euclidean distance was shorter to the jab and cross moves, however they should not belong to s0. There is potential to rectify this by assigning weights to each value of the vectors.

Further research may involve using a classifier to learn the likely transition functions, rather than the variance. This may lead to a more robust architecture, and one that is less prone to anomaly contamination.

REFERENCES

[1] S. Ortiz B, K. Moriyama, K. Fukui, S. Kurihara and M. Numao, "Three-Subagent Adapting Architecture for Fighting Videogames," *PRICAI 2010: Trends in Artificial Intelligence,* vol. 6230, B. Zhang and M. Orgun, Eds. Springer Berlin / Heidelberg, 2010, pp. 649-654.

[2] B. Cho, S. H. Jung, Y. R. Seong and H. R. Oh, "Exploiting intelligence in fighting action games using neural networks," *IEICE Trans. on Information and Systems*, vol. E89-D, no. 3,, 2006, pp. 1249–1256.

[3] S. Saini, P. W. H. Chung and C. W. Dawson, "Mimicking Human Strategies in Fighting Games using a Data Driven Finite State Machine," *2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference,* vol 2, P. Yan and B. Xu, Eds. Chongqing: IEEE Press, 2011, pp. 389-393.

[4] A. Fernández and S. Gómez, "Solving Non-uniqueness in Agglomerative Hierarchical Clustering Using Multidendrograms,," Journal of Classification, vol. 25, pp. 43-65, 2008.