Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2014)
Vladlen Koltun and Eftychios Sifakis (Editors)

SteerFit: Automated Parameter Fitting for Steering Algorithms Supplementary Document

Glen Berseth[1], Mubbasir Kapadia[2], Brandon Haworth[1] and Petros Faloutsos[1]

[1]York University
[2]University of Pennsylvania

1. Steering Algorithms

Our approach can be applied to any steering algorithm. For demonstration reasons, we use the following established algorithms that model different steering approaches.

1. **PPR**. [SKH*11] presents a hybrid framework that combines reaction, prediction and planning. It is an example of a rule based method for agent based steering and has 38 independent parameters. For example, *avoidance-turn-rate* defines the turning rate adjustment speed in proportion to the typical speed and *query-radius* controls the radius around an agent that **PPR** uses to predict collisions with other objects and agents.
2. **ORCA**. [vdBGLM11] is a very popular method that uses optimal reciprocal collision avoidance to efficiently steer agents in large-scale crowds. A subset of its independent parameters are: *max-neighbors*, the maximum number of nearby agents that an agent will take into consideration when making steering choices, *max-speed*, the maximum speed that an agent may travel with, and *time-horizon*, the minimal time for which an agent's computed velocity is safe with respect to other agents.
3. **SF**. [HFV00] uses hypothetical social forces for resolving collisions between interacting agents in dense crowds. In addition to general parameters similar with the other methods, each social force model has associated parameters that govern its relative influence. The effect of some of these parameters on the emergent dynamics of a crowd simulation has been studied before.

Table 2 reports the default, minimum, maximum, and optimal parameter values for **PPR** corresponding to each individual metric, as well as a uniform weighted combination of all metrics. Similarly, Table 3 reports the parameter values for **ORCA** and Table 4 reports the parameter values for **SF**.

2. Uni-Variate Parameter Analysis

This section describes a prefatory analysis we performed to understand the effect of the independent parameters on an algorithm's performance, and serves as a precursor to the multi-variate analysis reported in the paper. By varying each parameter in isolation and studying its effects on the performance criteria, we can answer questions such as: Which parameters are important? What are the bad values we need to avoid? Are the default values good?

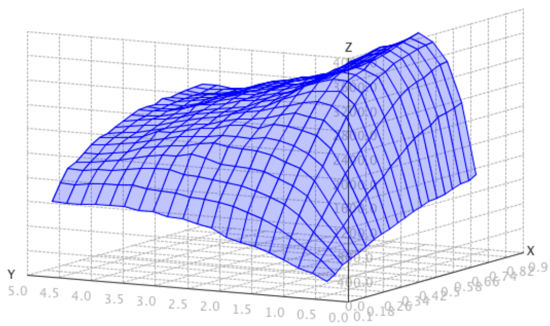


Figure 1: *The deficiency of* ***PPR****, $d(A_{\mathbf{v}}^{ppr})$ (vertical axis), with respect to two parameters, ped-avoid-rate and typical speed.*

For reference, we first compute the deficiency $d(A_{\mathbf{v}})$, distance quality $q^{\mathrm{d}}(A_{\mathbf{v}})$, and efficiency $e(A_{\mathbf{v}})$ metrics for the test set, $\mathcal{T}$, for the **PPR** algorithm using their default parameters, provided in Table 1. For default parameter settings, **PPR** can respectively solve 62% of the sampled scenarios.

To study the effect of each parameter in isolation, we sample each parameter of the steering algorithm independently in a bounded interval taking 20 uniformly distributed samples. The parameter bounds are chosen separately for each

Algorithm	v	$d(A_\mathbf{v})$	$q^{\mathrm{d}}(A_\mathbf{v})$	$e(A_\mathbf{v})$	$f(A_\mathbf{v})$
PPR	DEF	0.39	0.49	0.96	0.61
	UNI	0.25	0.25	0.95	0.46

Table 1: *Comparison of $d(A_\mathbf{v})$,$q^{\mathrm{d}}(A_\mathbf{v})$, $e_s(A_\mathbf{v})$, and $f(A_\mathbf{v})$ wich is equal to the equally weighted combination of the 3 metrics for the **PPR** steering algorithm using: (a) `DEF`: default parameter values and (b) `UNI`: parameter values obtained using uni-variate analysis.*

parameter based on intuition, physical interpretation of the parameter, or default values provided by the algorithm's creators. Table 2 enumerates the bounds of the parameters for **PPR**.

We find that the deficiency of **PPR** is sensitive to 23 of its 38 parameters. For each parameter we can also identify its optimal value and make trade offs. Table 1 shows the maximum improvement in the value of the performance metrics that we can achieve using this analysis (labelled UNI in the table) compared to value of the metrics that correspond to the default values (labelled DEF in the table).

The deficiency for **PPR**, $d(A_\mathbf{v}^{ppr})$, decreases to 25% by selecting the optimal values of ped-avoid-rate and typical speed. The quality with respect to distance travelled for **PPR**, $q^{\mathrm{d}}(A_\mathbf{v}^{ppr})$, decreases to 0.25 for the optimal values of ped-avoid-rate and typical speed.

Efficiency is an important issue for steering algorithms. We can see that, as expected, the e of **PPR** decreases with the query radius. However, the more interesting observation comes from Table 1, where we can see that the e metric for the **PPR** algorithm improves significantly when we use the appropriate parameter values from this analysis.

Optimizing for a weighted combination of all three metrics yields also interesting results. We observe that ped-avoid-rate $= 0.55$ produces optimal results in the **PPR** algorithm for an equal proportion of the 3 $d(A_\mathbf{v}^{ppr})$, $q^{\mathrm{d}}(A_\mathbf{v}^{ppr})$ and $e_s(A_\mathbf{v}^{ppr})$.

Knowing how each parameter affects each performance metric, allows us to potentially focus our optimization efforts on specific parameters based on the requirements of an application. For example, we found that ped-avoid-rate has little effect on efficiency, $e(A_\mathbf{v}^{ppr})$ while it does affect deficiency, $d(A_\mathbf{v}^{ppr})$, and quality, $q^{\mathrm{d}}(A_\mathbf{v}^{ppr})$. Therefore, it may be a suitable parameter to explore if we need to improve quality or deficiency without affecting efficiency.

To gain insight on the simultaneous effect of multiple variables we perform one bi-variate analysis for **PPR**. Figure 1 shows the deficiency of **PPR** with respect to the Cartesian product of two parameters, ped-avoid-rate, and typical speed. The shape of the resulting surface indicates that deficiency depends non-trivially on both parameters at the same time.

Discussion. The analysis in this section offers valuable insights on the effects of each parameter on the objectives.

- We can easily identify which values of the parameters we should avoid, and which might be good choices.
- The experiments indicate that for an algorithm the default parameters are not necessarily optimal. They also verify that, as expected, the objectives are generally not separable functions of the parameters, v. We therefore need to fit the parameters simultaneously using a multi-variate optimization method.
- For **PPR** we might be able reduced the number of parameters that we need to fit from 38 to the 23 that seem more important, which may significantly improve the time it takes to perform optimal fitting.

3. Algorithmic Details for CMA-ES

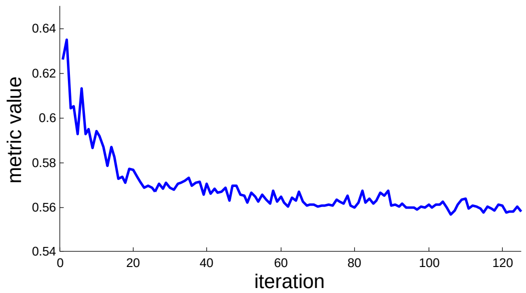
Algorithm 2 describes the details of the CMA-ES algorithm we used for automatically selecting parameter values that optimize a given objective function.

The CMA-ES algorithm searches iteratively the parameter space for the optimal parameter values in an evolutionary fashion. At each iteration it generates N-samples of the parameter vector and keeps a subset of the samples that exhibit high fitness (minimize the objective). The algorithm then tries to increase the probability of successful candidate solutions and search steps, in a maximum-likelihood sense. The mean of the probability distribution of the samples is updated such that the likelihood of successful solutions is increased. A covariance matrix that captures the pair-wise dependencies between parameter distributions is also updated such that the likelihood of previously successful steps is increased. Samples are taken from a normal multivariate distribution with the computed mean and covariance matrix. A key feature of the algorithm is the way it controls the step size between iterations and the evolution paths. For more details see [HO96], and http://en.wikipedia.org/wiki/CMA-ES.

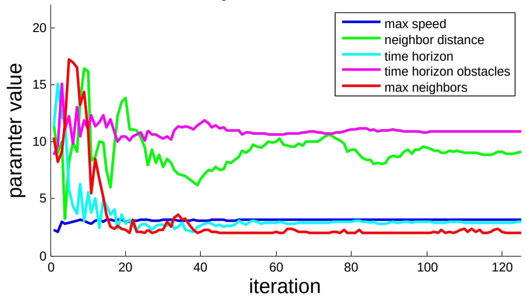
`input` Test set $\mathcal{T}$, Objective $f(A_\mathbf{v}, w)$, Algorithm $A_\mathbf{v}$, parameters $\mathbf{v} \in \mathbb{V}$
Initialize, mean m, covariance matrix C
while *not* `termination_condition` **do**
    while $i < N$ **do**
        $\mathbf{v}_i$ = `Sample` $\mathcal{N}(m, C)$
        `Compute Objective` $f_i = f(A_{\mathbf{v}_i}, w)$
    end while
    $\{\mathbf{v}_0, \mathbf{v}_1 ... \mathbf{v}_{N-1}\} = \arg\mathrm{sort}_{\{\mathbf{v}_i\}}(\{f_i | \forall i\})$
    $\mathbf{v}^*$ = `Update` $(\{\mathbf{v}_0, \mathbf{v}_1 \cdots \mathbf{v}_{N-1}\})$
    `Update Mean`, m
    `Update search paths`
    `Update Covariance Matrix`, C
end while
return $\mathbf{v}^*$

Figure 2: *Main loop of CMA-ES Algorithm for parameter optimization of steering algorithms.*

Example: Figure 3 illustrates an optimization process. The parameters of **ORCA** $\mathbf{v}$ = {max speed, neighbour distance, time horizon, time horizon obstacles, max neighbours} are optimally fitted to an equally weighted combination of metrics over the test set $\mathcal{T}$. After 45 iterations the optimization converges to approximately 10% better objective value. A quick observation shows that the optimization has reduced the number of neighbours that the algorithm considers for each agent, max neighbours, from 10 to 2.



(a) Objective Function



(b) **ORCA** parameter values

Figure 3: *Optimizing **ORCA** parameters to minimize the uniformly weighted combination of metrics over the test set $\mathcal{T}$. Each iteration is equal to 8 metric evaluations. As can be seen convergence occurs around 45 iterations.*

4. Parameter Optimization Validation

We validate the effectiveness of the parameter optimization by calculating the metric values over a different set of test cases. The results of this can be seen in Table 5 where it can be see that for all metrics and all algorithms there is a similar improvement over the default values.

5. Parameter and Metric Analysis

In our analysis we also computed correlations between the parameters of the steering algorithms and the metrics. Tables 8, 6, 7 list the results of this analysis.

6. More Applications and Results

Dynamically Adapting Steering Parameters. Our methodology can create a large number of samples that relate parameter values to performance metrics. It is an obvious next step to model that landscape, and learn the relation between parameters and performance metrics, using, perhaps, non-linear, manifold learning techniques. Figure 4 shows a snapshot from an interactive demo that allows the user to switch dynamically between optimal parameter values that correspond to different objectives.

$A_\mathbf{v}$	$\mathbf{v}$	q^e	d	q^d	e	q^t	u
PPR	DEF	0.53	0.39	0.49	0.96	0.57	0.59
	OPT	0.30	0.10	0.22	0.91	0.07	0.34
ORCA	DEF	0.67	0.53	0.61	0.84	0.56	0.64
	OPT	0.62	0.51	0.57	0.82	0.29	0.58
SF	DEF	0.46	0.27	0.42	0.89	0.50	0.51
	OPT	0.23	0.05	0.20	0.81	0.30	0.33

Table 5: *Validation of the Comparison of $q^e(A_\mathbf{v})$, $d(A_\mathbf{v})$, $q^d(A_\mathbf{v})$, $e_s(A_\mathbf{v})$, $q^t(A_\mathbf{v})$, and a uniform combination of all metrics for both steering algorithms using: (a) `DEF`: default parameter values and (b) `OPT`: optimal parameter values on second set of scenarios that were not used in training.*

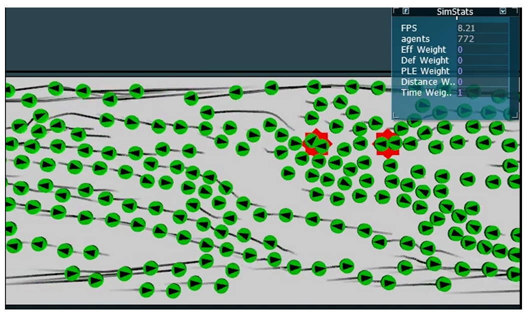


Figure 4: *A prototype system for interactively setting the relative weights of the metrics in the objective. When the weights change, the algorithm's parameters are set automatically to the corresponding optimal values.*

Tables 9 and 10 contain the default and optimized objective values for the three steering algorithms.

References

[HFV00] Helbing D., Farkas I., Vicsek T.: Simulating dynamical features of escape panic. *Nature 407*, 6803 (2000), 487–490. 1

[HO96] Hansen N., Ostermeier A.: Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *IEEE International Conference on Evolutionary Computation* (1996), pp. 312–317. 2

[SKH*11] Singh S., Kapadia M., Hewlett B., Reinman G., Faloutsos P.: A modular framework for adaptive agent-based steering. In *ACM I3D* (2011), pp. 141–150. 1

[vdBGLM11] van den Berg J., Guy S. J., Lin M., Manocha D.: Reciprocal n-body collision avoidance. In *Robotics Research*, vol. 70. 2011, pp. 3–19. 1

$A_{\mathbf{v}}$	**v**	d	q^{d}	q^{t}	q^{e}	e	u
PPR	DEF	0.39	0.49	0.56	0.53	0.96	0.58
	OPT	0.09	0.20	0.07	0.28	0.89	0.34
ORCA	DEF	0.56	0.61	0.56	0.67	0.75	0.62
	OPT	0.47	0.56	0.30	0.63	0.67	0.55
SF	DEF	0.26	0.41	0.50	0.45	0.87	0.50
	OPT	0.04	0.20	0.29	0.23	0.78	0.32

Table 9: *Comparison of failure rate d, distance quality q^{d}, time quality q^{t}, effort quality q^{e}, computational efficiency e, and a uniform combination of all metrics u for the three steering algorithms using: (a) `DEF`: default parameter values and (b) `OPT`: optimal parameter values.*

$A_{\mathbf{v}}$	**v**	2-agent-crossing	two-way hallway
PPR	DEF	3.42	3.40
	OPT	1.92	2.27
ORCA	DEF	2.12	2.95
	OPT	0.63	2.20
SF	DEF	3.74	3.62
	OPT	3.10	2.76

Table 10: *Comparison of entropy metric values before and after optimization to match real world data. `DEF`: default parameter values, `OPT`: optimal parameter values.*

Parameter Name	DEF	Min	Max	d	q^{d}	q^{t}	q^{e}	e	u
max speed	2.60	1	4	3.29	1	4.00	1.66	4	3.03
max force	14	8	22	14.45	15.15	22	18.76	15.62	19.50
max speed factor	1.70	0.60	4.70	3.38	1.11	4.70	0.60	3.77	3.45
faster speed factor	1.31	0.55	4.20	0.62	3.92	0.81	2.84	1.22	0.71
slightly faster speed factor	1.15	0.40	3.40	1.72	2.41	3.40	3.09	0.90	2.29
typical speed factor	1	0.50	1.50	0.53	0.50	1.50	1.04	0.50	1.50
slightly slower speed factor	0.77	0.15	1.20	0.22	0.19	1.20	0.70	0.59	0.84
slower speed factor	0.50	0.10	1	0.11	0.10	0.10	0.10	0.57	0.10
cornering turn rate	1.90	0.83	3.76	3.45	2.30	1.69	3.51	2.64	1.53
adjustment turn rate	0.16	0.03	1.54	0.03	0.58	0.29	0.13	0.29	0.37
faster avoidance turn rate	0.55	0.15	1.87	0.72	1.06	1.01	0.79	0.89	1.30
typical avoidance turn rate	0.26	0.08	0.75	0.66	0.62	0.75	0.59	0.62	0.71
braking rate	0.95	0.50	1.50	0.52	1.50	0.55	1.17	0.67	1.44
comfort zone	1.50	0.70	2.80	1.41	1.70	1.32	2.01	0.86	1.63
query radius	10	5	21	17.40	11.03	5	8.22	5	5
similar direction threshold	0.94	0.78	1.00	0.93	0.95	0.78	0.89	0.99	0.80
same direction threshold	0.99	0.89	1.00	0.90	0.89	0.91	0.92	0.91	0.93
oncoming prediction threshold	−0.95	−0.99	−0.78	−0.97	−0.81	−0.81	−0.92	−0.99	−0.92
oncoming reaction threshold	−0.95	−0.99	−0.78	−0.87	−0.85	−0.78	−0.94	−0.95	−0.87
wrong direction threshold	0.55	0.23	0.78	0.26	0.23	0.23	0.29	0.45	0.25
threat distance threshold	8	3	16.80	13.70	16.19	8.59	6.02	6.90	9.48
threat min time threshold	0.80	0.37	1.45	0.38	0.79	1.17	0.39	1.11	0.37
threat max time threshold	4	1.22	8.77	7.99	5.15	6.46	8.21	3.97	3.99
predictive anticipation factor	5	2.33	8.39	4.30	4.74	4.78	6.87	5.85	5.38
reactive anticipation factor	1.10	0.33	2.31	0.95	1.03	0.97	1.01	0.62	0.99
crowd influence factor	0.30	0.11	0.61	0.35	0.22	0.30	0.44	0.11	0.59
facing static object threshold	0.30	0.08	0.61	0.09	0.34	0.29	0.61	0.19	0.46
ordinary steering strength	0.05	0.00	0.20	0.02	0.00	0.00	0.08	0.11	0.02
oncoming threat avoidance strength	0.15	0.05	0.40	0.40	0.12	0.06	0.09	0.17	0.08
cross threat avoidance strength	0.90	0.73	1.00	0.76	0.91	0.95	0.74	0.90	0.95
max turning rate	0.10	0.02	0.23	0.10	0.10	0.15	0.13	0.10	0.10
feeling crowded threshold	3	1	8	2	2	1	4.06	1	5
scoot rate	0.40	0.17	0.78	0.78	0.60	0.78	0.78	0.72	0.71
reached target distance threshold	0.50	0.10	0.90	0.78	0.90	0.90	0.90	0.12	0.89
dynamic collision padding	0.20	0.02	0.43	0.43	0.24	0.17	0.20	0.16	0.19
furthest local target distance	20	10	50	34	22	39	15	10	10
next waypoint distance	50	30	70	62	39	64	38	32	44
max num waypoints	20	10	50	22	15	10	44	32	13

Table 2: *Parameters for **PPR** algorithm with their default values, bounds, and optimal values obtained using multi-variate analysis for different objective functions.*

Parameter Name	DEF	Min	Max	d	q^{d}	q^{t}	q^{e}	e	u
max speed	2	1	3.20	3.20	2.15	3.20	1.52	3.14	3.14
neighbor distance	15	2	22	17.39	13.37	14.75	12.08	8.18	8.99
time horizon	10	2	16	16	3.71	2	2.72	8.44	2.92
time horizon obstacles	7	2	16	12.30	16	9.60	11.81	2	10.92
max neighbors	10	2	22	8	11	2	15.03	2	2

Table 3: *Parameters for **ORCA** algorithm with their default values, bounds, and optimal values obtained for each metric separately, and a uniform combination of the metrics.*

Parameter Name	DEF	Min	Max	d	q^{d}	q^{t}	q^{e}	e	u
acceleration	0.50	0.05	2	0.05	0.05	0.05	0.05	1.90	0.05
personal space threshold	0.30	0.10	1	0.69	0.28	0.50	0.10	0.10	0.41
agent repulsion importance	0.30	0.05	1	0.05	0.05	0.05	0.11	0.66	0.38
query radius	4	1	10	1	10	9.44	10	2.08	3.28
body force	1500	500	5000	2431.40	2778.10	3832.20	500	3498.40	4858.80
agent body force	1500	500	5000	500	4677.80	1573.70	4027.40	3009.50	1073.20
sliding friction force	3000	1000	10000	3281.10	1000	6795.70	10000	8489.20	6091.30
agent b	0.08	0.01	5	0.09	0.08	0.09	0.11	3.81	0.13
agent a	25	1	100	46.25	48.21	58.27	53.24	52.00	53.37
wall b	0.08	0.01	5	0.15	0.10	0.18	0.08	5	0.09
wall a	25	1	100	100	67.15	55.05	61.65	98.20	60.87

Table 4: *Parameters for* ***SF*** *algorithm with their default values, bounds, and optimal values obtained using multi-variate analysis for different objective functions.*

Parameter	$d(A_{\mathbf{v}})$	$q^{\mathrm{d}}(A_{\mathbf{v}})$	$q^{\mathrm{t}}(A_{\mathbf{v}})$	$q^{\mathrm{e}}(A_{\mathbf{v}})$	$e(A_{\mathbf{v}})$
max speed	0.02	0.03	−0.34	0.58	0.14
neighbour distance	−0.09	−0.07	−0.13	−0.03	0.03
time horizon	−0.12	−0.08	0.10	0.04	0.07
time horizon obstacles	−0.09	−0.09	0.17	0.04	0.11
max neighbors	0.42	0.47	0.54	0.29	0.37

Table 6: *This tables shows Spearman rank correlation coefficients between 5 metrics and all of the parameters for the* ***ORCA*** *algorithm.*

Parameter	$d(A_{\mathbf{v}})$	$q^{\mathrm{d}}(A_{\mathbf{v}})$	$q^{\mathrm{t}}(A_{\mathbf{v}})$	$q^{\mathrm{e}}(A_{\mathbf{v}})$	$e(A_{\mathbf{v}})$
acceleration	0.14	0.18	0.15	0.18	−0.17
personal space threshold	−0.02	−0.01	−0.01	−0.01	0.02
agent repulsion importance	0.04	0.04	0.04	0.04	0.04
query radius	−0.01	−0.01	−0.01	−0.01	−0.00
body force	0.05	0.05	0.04	0.05	0.04
agent body force	0.00	0.01	0.00	0.01	−0.02
sliding friction force	0.00	0.01	0.01	0.01	−0.01
agent b	0.02	0.14	0.15	0.13	−0.37
agent a	−0.27	−0.21	−0.24	−0.21	−0.25
wall b	0.66	0.65	0.62	0.66	−0.01
wall a	0.37	0.37	0.34	0.37	−0.04

Table 7: *This tables shows Spearman rank correlation coefficients between 5 metrics and all of the parameters for the* ***SF*** *algorithm*

Parameter	$d(A_{\mathbf{v}})$	$q^{\mathrm{d}}(A_{\mathbf{v}})$	$q^{\mathrm{t}}(A_{\mathbf{v}})$	$q^{\mathrm{e}}(A_{\mathbf{v}})$	$e(A_{\mathbf{v}})$
max speed	−0.06	−0.12	−0.24	−0.04	−0.04
max force	−0.40	−0.41	−0.45	−0.38	−0.13
max speed factor	−0.58	−0.63	−0.72	−0.57	−0.23
faster speed factor	0.35	0.34	0.33	0.32	0.23
slightly faster speed factor	−0.06	−0.12	−0.25	−0.08	−0.06
typical speed factor	−0.40	−0.43	−0.62	−0.28	−0.26
slightly slower speed factor	0.30	0.28	0.28	0.26	0.00
slower speed factor	0.30	0.27	0.16	0.25	0.06
cornering turn rate	0.15	0.08	0.07	0.13	0.18
adjustment turn rate	−0.21	−0.24	−0.23	−0.22	−0.18
faster avoidance turn rate	−0.39	−0.39	−0.39	−0.35	−0.19
typical avoidance turn rate	−0.33	−0.34	−0.39	−0.37	−0.27
braking rate	−0.32	−0.28	−0.26	−0.27	−0.12
comfort zone	−0.30	−0.26	−0.26	−0.23	0.02
query radius	0.29	0.33	0.38	0.34	0.63
similar direction threshold	0.15	0.11	0.11	0.14	0.14
same direction threshold	0.52	0.55	0.64	0.52	0.11
oncoming prediction threshold	0.03	0.02	0.04	0.05	0.13
oncoming reaction threshold	−0.48	−0.50	−0.58	−0.49	−0.25
wrong direction threshold	0.23	0.25	0.29	0.23	0.05
threat distance threshold	0.12	0.10	0.14	0.13	0.00
threat min time threshold	0.38	0.40	0.46	0.37	0.19
threat max time threshold	−0.01	−0.04	−0.07	−0.00	0.02
predictive anticipation factor	−0.30	−0.29	−0.27	−0.28	−0.21
reactive anticipation factor	0.01	0.02	0.12	0.13	0.05
crowd influence factor	−0.35	−0.35	−0.38	−0.31	−0.12
facing static object threshold	0.21	0.21	0.27	0.18	−0.05
ordinary steering strength	0.04	0.03	0.07	0.02	0.04
oncoming threat avoidance strength	−0.25	−0.31	−0.35	−0.23	−0.16
cross threat avoidance strength	−0.08	−0.12	−0.18	−0.14	−0.01
max turning rate	0.43	0.35	0.33	0.29	0.17
feeling crowded threshold	−0.49	−0.53	−0.56	−0.46	−0.30
scoot rate	−0.12	−0.17	−0.24	−0.17	−0.11
reached target distance threshold	−0.26	−0.41	−0.44	−0.36	−0.30
dynamic collision padding	0.15	0.15	0.25	0.18	0.11
furthest local target distance	0.16	0.19	0.25	0.17	0.65
next waypoint distance	−0.07	−0.04	0.07	−0.07	0.01
max num waypoints	0.39	0.41	0.43	0.35	0.14

Table 8: *This tables shows Spearman rank correlation coefficients between 5 metrics and all of the parameters for the* ***PPR*** *algorithm*