Undefined 0 (2016) 1–0
IOS Press

A Study on Performance Metrics and Clustering Methods for Analyzing Behavior in UAV Operations

Víctor Rodríguez-Fernández [a,*], Héctor D. Menéndez [b] and David Camacho [a]
[a] *Universidad Autónoma de Madrid (UAM) 28049, Madrid, Spain*
E-mail: victor.rodriguez@inv.uam.es ,david.camacho@uam.es
[b] *University College London (UCL), London, UK*
E-mail: h.menendez@ucl.ac.uk

Abstract. Unmanned Aerial Vehicles (UAVs) are starting to provide new possibilities to human societies and their demand is growing according to the new industrial application fields for these revolutionary tools. The current systems are still evolving, specially from an Artificial Intelligence perspective, which is increasing the different tasks that UAVs can perform. However, the current state still requires a strong human supervision. As a consequence, a good preparation for UAV operators is mandatory due to some of their applications might affect human safety. During the training process, it is important to measure the performance of these operators according to different factors that can help to decide what operators are more suitable for different kinds of missions creating operator profiles. Having this goal in mind, this work aims to present an extensive and robust methodology to automatically extract different performance profiles from the training process of operators in an UAV simulation environment. Our method combines the definition of a set of performance metrics with clustering techniques to define operators profiles, ensuring that the behavior discrimination is suitable and consistent.

Keywords: UAVs, Human-Robot Interaction, Computer-based Simulation, Clustering, Performance metrics, Behavioral analysis

1. Introduction

Managing Unmanned Aerial Vehicles (UAVs) is a sensitive task which requires a strong preparation, specially now that theses systems are evolving to cooperative environments were multiple UAVs try to perform a specific task [28].

The work of UAV operators is extremely critical due to the high costs involving any UAV mission, both financial and human. Thus, lot of research in the field of human factors, and more specifically, in Human Supervisory Control (HSC) and Human-Robot Interaction (HRI) systems, have been carried out, in order to understand and improve the performance of these operators [24].

In recent years, two topics are emerging in relation to the study of Unmanned Aircraft Systems (UASs). One is the effort to design systems such that the current many-to-one ratio of operators to vehicles can be inverted, so that a single operator can control multiple UAVs. The other is related to the fact that accelerated UAS evolution has now outpaced current operator training regimens, leading to a shortage of qualified UAS pilots. Due to this, it is necessary to re-design the current intensive training process to meet that demand, making the UAV operations more accessible and available for a less limited pool of individuals, which may include, for example, video-game players [26].

This work is focused on measuring and analyzing the performance of inexperienced UAV operators using the data extracted from a multi-UAV simulator. This performance data will be used to extract behav-

*Corresponding author. E-mail: victor.rodriguez@inv.uam.es.

ioral patterns among users, which could be used to select potential UAV operators. The main contributions of this paper over the previous work [31] can be summarized as follows:

- We extend the related work and backgrounds about the problem domain (See section 2).
- We extend the metrics used in our previous work, combining them with similar metrics extracted from [32], and we introduce the Reflexes as a new metric to complement some behavioural features that were not considered during the previous works. Furthermore, the relationships among these metrics are evaluated to check whether they complement each other. The metrics are presented in Section 4.1 and they are evaluated in Section 6.1.
- The dataset has been incremented a 50% from the original one (see Section 5). This will help to improve the generalization of the model. Furthermore, we also increment the number of different clustering techniques applied, in order to improve the clustering validation process (see Section 5)
- We have extended the clustering validation process dividing the validation measures according to two main goals: the clear discrimination among the clusters (quality) and the clusters robustness with respect to the chosen features (stability). The selection of the final clustering solution is described in Section 4.2 and the experimental application in Section 6.2.

The rest of the paper is structured as follows: next Section provides a discussion about the related work. Section 3 gives a brief review of the simulation environment used to extract the data for this work. Section 4 details all the process for creating performance profiles to analyze the behavior of operators. After, Section 5 introduces the research questions and the experimental setup. The results of these experiments are discussed in Section 6 using different clustering techniques and validation metrics. Finally, Section 7 presents the conclusions and future work.

2. Related Work

This section aims to describe the related work around current research of UASs, specially focused on those techniques used to measure operators performance. We also introduce the learning techniques applied in this work for behavioural model analysis, including specific quality metrics that will be used to evaluate these models.

2.1. Unmanned Aerial Vehicles

UAVs are becoming a relevant area with several applications for human societies. These systems have been used in a wide range of fields, where the most remarkable are: agriculture [18], rescue [16], architecture [10] and traffic monitoring [12] among others.

Research in this area, from a Computer Science perspective, aims to ensure the managing and security of these devices, keeping their autonomy and efficiency. The main factors under study are specially focused on path and mission planning strategies, [29,27] cooperation for multi-UAVs scenarios [4,5] and operators training [31], among others.

With regard to the fields of mission planning and path planning, recent works aim to define fast strategies during the decision process. For example, Ramírez-Atenza et al. [29] defines the problem of mission planning as a Temporal Constraints Satisfaction Problems (TCSP). From the path planning perspective, we can also find search based approaches for the path optimization problem. For example, Nikolos et al. [27] define an evolutionary algorithm for UAV navigation environments where the algorithm is able to produce an optimal path offline and online.

The definition of cooperation strategies has been usually assigned from a multi-agents perspective, where different game theory based models have been applied for multi-learning task and multi-cooperation. The works of Beard and McLain are good examples about how the topology for the cooperation systems can be defined [4] and how practical problems, such as collisions, can be avoid when there are strong communication constrains [5].

Finally, from the operator's training perspective, we can find some interesting works developed by McCarley and Wickens where they try to measure the implication of introducing human factors in the UAV managing process [25], specially focused on pilot-based interfaces and air traffic management. From this perspective, Ayaz et al. studied how the mental conditions can be monitored during the training process in order to ensure that the trainee can handle strong workloads during the managing process [2]. Our previous work targets this area [31], aiming to describe different behavioural profiles with which the trainees can be grouped.

2.2. Performance Analysis of UAV Operators

The most common metrics to assess human performance on HRI systems focus on the operator workload and its *Situational Awareness* [14]. However, it is also interesting to define some metrics that collect the performance of an operator in a direct way, as a global *score* indicating the performance quality. These metrics, also known as *Direct measures of performance quality*, create a *user profile*, and are widely used, for example, in the world of videogames [6].

The information given by the different metrics and operator interactions can help to recognize and extract some hidden information about the general use of the system. Here, *data mining and machine learning* techniques take much importance. Since multi-UAV systems are still futurist developments, it is difficult and costly to find experts able to label the operator interactions in order to make an objective supervised analysis, hence we can only work in this field by using unsupervised learning techniques [9].

For this reason, the analysis made in this work is focused on *Clustering*, a popular unsupervised technique used to group together, in a blindly way, objects which are similar to one another.

2.3. Clustering

Clustering is a learning technique which is normally categorized within unsupervised learning, due to its ability to detect patterns in data blindly [22]. These algorithms have been applied to an extensive range of fields, including biology [21], text mining [7], feature selection [19] and security [3], among others.

These techniques are based on grouping the data according to some similarity criteria. Usually, clustering algorithms are categorized into three main areas: partitional clustering [23] (where the algorithm divides the dataset into disjoint sets), overlapping clustering [1] (where one or more data instances can belong to none, one or several clusters) and hierarchical [20] (where the data are nested in hierarchical levels).

The range of clustering algorithms is extremely big and new algorithms appear very frequently. In this work we are concerned with the following algorithms, which are well-known in the community: K-means [23], PAM (Partition Around Medoids) [20], SOM (Self-Organizing Maps) [21], AGNES (AGglomerative NESting) [20] and SOTA (Self-Organizing Tree Algorithm) [17].

One of the most relevant problems in clustering is to choose a proper algorithm and a good number of clusters for a specific application. There are several different ways for making this decision, but all of them depend on the final research goal. In this area, we specially focus on two main criteria: the internal validation of the clusters (here called *quality*), which ensures that the final clusters have a strong cohesion and discrimination, and the robustness of the clustering (or *stability*), which measures the ability of the solutions to be stable even when some features of the space are unknown. Good examples of internal validation metrics are the *Dunn Index* [15], *Silhouette Width* [33] and the *Calinski and Harabasz index (CH)* [11].

The stability measures are a special version of internal measures. They evaluate the consistency of a clustering result by comparing it with the cluster obtained after each column of the data matrix is removed, one at a time. Well-known stability measures are: Average proportion of non-overlap (APN) and *Average Distance* (AD), both of them described in [13].

3. DWR - A Multi-UAV Simulation Environment

Retrieving data from the interactions and performance of UAV operators during a multi-UAV mission simulation is a novel task, due to the premature state of this field. This is causing an impediment to expand the analysis to an accessible place, where an inexpert user could be trained to become a potential expert in UAV operations [26].

For this reason, the simulation environment used as the basis for this work has been designed following the criteria of accessibility and usability. It is called Drone Watch And Rescue (DWR) [1], and its complete description can be found in [30]. DWR gamifies the concept of a multi-UAV mission (See Figure 1), challenging the operator to capture all mission targets consuming the minimum amount of resources, while avoiding at the same time the possible incidents that may occur during a mission (e.g., Danger Areas, Sensor breakdowns). To avoid these incidents, an operator in DWR can perform multiple interactions to alter both the UAVs in the mission and the waypoints composing their mission plan.

Besides, it is remarkable how DWR saves data from a simulation. Whenever an event occurs during a simulation, DWR stores the simulation status in that mo-

[1] Available at http://savier.ii.uam.es:8888/#/home

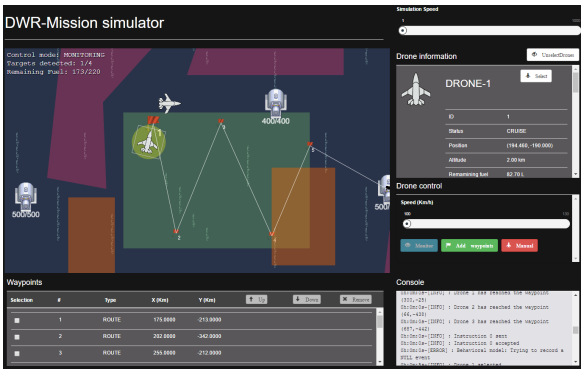

Fig. 1. Screenshot of Drone Watch And Rescue (DWR).

ment, as a *Simulation Snapshot*. This snapshot contains all relevant information of the current status of every element taking part in the simulation. The information stored by DWR allows to reproduce the entire simulation, which is helpful for the analysis process.

4. Behavioral analysis through performance metrics and clustering methods

This section describes the different techniques developed in this work for analyzing the behavior of operators in the environment DWR, based on their performance during simulations. Below are described the two main blocks comprising this analysis, namely, the definitions of performance metrics and the validation of an optimal discrimination to distinguish shared performance profiles.

4.1. Direct measures of performance quality

The main goal of this work leads us to the need for defining a way to measure the performance of a user in a specific simulation.

To achieve this, seven performance metrics have been defined: Agility (AG), Consumption (C), Score (S), Attention (AT), Aggressiveness (AGR), Precision (P) and Reflexes(R). All of them are numeric values in the range $[0, 1]$, where 0 represents the worst performance for that metric, and 1 represents the best.

Each of these metrics are computed for a specific simulation. Based on that, we can define, for a given user, his *performance profile* as the tuple $(AG, C, S, AT, P, AGR, R)$. Below the seven metrics used in this work are described.

4.1.1. Agility

Agility (*AG*) measures the average speed with which the user has interacted with the simulator. In the simulation environment, a user can manipulate the simulation speed, giving values from 1 to 1000 times. A user is considered agile if he can interact when things are happening fast. Let $I(s)$ be the set of all interactions performed during a given simulation s, the Agility is computed as:

$$A(s) = \frac{\sum_{i \in I(s)} \frac{simulationSpeed(i)}{MAX\_SPEED}}{|I(s)|} \quad (1)$$

where $simulationSpeed(i)$ and $MAX\_SPEED = 1000$ gives the speed in which the simulation was running at the moment when the interaction i was made.

4.1.2. Consumption

Consumption (*C*) metric measures the fuel consumed throughout the simulation time. Given a specific instant (also called *snapshot*) sh of a simulation s, we can compute the global remaining fuel (rf) at that instant as

$$rf(s, sh) = \sum_{u \in U(s)} rf(u, sh) + \sum_{r \in R(s)} rf(r, sh) \quad ,$$

where $U(s)$ is the set of UAVs participating in the simulation s and $R(s)$ is the set of refueling stations taking part in the Mission Scenario of simulation s (See DWR description in [30]). When a UAV u is destroyed during the simulation, it is considered that $rf(u, sh) = 0$ for every instant sh after the UAV destruction.

To calculate the consumption over a simulation s, we compare the remaining fuel at the end of the simulation (last snapshot, or *lSh*) with that at the beginning (first snapshot, or *fSh*):

$$C(s) = \frac{rf(s, lSh(s))}{rf(s, fSh(s))} \quad (2)$$

High values of this metric indicate that the remaining fuel at the end of the mission is high, so the consumption is considered low. On the other hand, low values mean high consumption rate. This metric also gives information about the duration of a simulation: since a user in DWR can abort a mission whenever he wants, high values of consumption will likely be associated to short missions, while low values will indicate long ones.

4.1.3. Score

The *Score* (*S*) metric gives a global success/failure rate of a simulation. The main goal for an operator monitoring a simulation in DWR is to capture the maximum number of targets, minimizing the resources consumed. This goal can be divided into 2 sub-goals: detecting targets and minimizing UAV.

Based on this description, we define the score of a simulation s as:

$$S(s) = \frac{1}{2}\left[\frac{|tD(s)|}{|T(s)|} + \left(1 - \frac{|dUAVs(s)|}{|U(s)|}\right)\right] \quad (3)$$

where $tD(s)$ and $dUAVs(s)$ refer to the targets detected and the UAVs destroyed respectively up to time t, $T(s)$ is the set of all mission targets and $U(s)$ is the set of all UAVs participating in the mission.

4.1.4. Attention

The *Attention* (*AT*) metric rates the user intensity in terms of the interactions he has performed in a simulation. Given a simulation s, Attention is defined as:

$$AT(s) = 1 - \frac{1}{1 + \sqrt{|I(s)|}}, \quad (4)$$

where $I(s)$ is the set of all interactions performed during simulation s.

4.1.5. Aggressiveness

One of the most important features according to the operator attitude is related to his *Aggressiveness* (*AGR*). In order to measure this value, we have defined it based on how the operator modifies the paths of the UAVs in the mission. The main control modes for performing path operations are: the *Monitor* mode, the *Add waypoints* mode and the *Manual* mode. The first mode allows the user to move a waypoint, the second is used to include or delete waypoints to the current UAV path, and the last is used to generate a totally new path. In this work, an aggressive behaviour is considered when the operator eliminates the whole path and creates a new one , i.e., *Manual* mode would be the most aggressive, followed by *Add waypoints* and *Monitor*.

Since we will measure the Aggressiveness according to the waypoint modifications in the three different modes, we define the sets $W_{MO}(s)$, $W_A(s)$ and $W_{MA}(s)$ which represent the set of interactions with waypoints performed during the *Monitor*, *Add waypoints* and *Manual* mode, respectively. Using these parameters, the metric is defined as:

$$A(s) = \frac{\alpha|W_{MA}(s)| + \beta|W_A(s)| + \gamma|W_{MO}(s)|}{|W(s)|},$$

where $W(s) = W_{MO}(s) \cup W_A(s) \cup W_{MA}(s)$. The values α, β, γ are weighted coefficients used for balancing the aggressive factor of each type of interaction ($\alpha, \beta, \gamma < 1, \quad \alpha > \beta > \gamma$,). When this metric achieves the maximum value, it means that the user has featured an aggressive behaviour, strongly modifying the UAV path. Otherwise, when the value tends to 0, it indicates that he has softly modified the waypoints.

4.1.6. Precision

The *Precision* (*P*) metric measures the replanning skills of a user on a simulation, rating how he has reacted to the mission alerts. The design of this metric is based in the following assumption: a precise operator should only perform replanning interactions (add/edit/remove waypoints) when an alert occurs. Therefore, the waypoints added when no alert has happened should penalize the precision rate. Based on this, we can divide the precision computation into two parts: The precision in times of alerts (*Alert Precision*, P_A) and the precision when nothing is altering the simulation, i.e, the operator must only monitor the simulation status (*Monitoring Precision*, P_M).

$$P(s) = \frac{P_A + P_M}{2} \quad (5)$$

The *Alert Precision* (P_A) supposes that every waypoint added/edited/removed during a specific interval time (10 seconds for this experiment) since the beginning of an alert is placed in order to avoid that incident, so it is considered as a precise interaction. Let $A(s)$ be the set of alerts happened during the simulation s, we can compute P_A as follows:

$$P_A(s) = \frac{\sum_{a \in A(s)} p_A(i, s)}{|A(s)|}$$

$$p_A(a, s) = 1 - \frac{1}{1 + |W_a(s)|},$$

where $p_A(a, s)$ gives the precision for an specific alert a. In this last equation, $W_a(s)$ is the set of all *waypoint interactions* (add/edit/remove) performed since the alert a started until 15 seconds after (i.e, interactions within the interval $[t(a), t(a) + 15]$, where $t(a)$

is the alert triggering timestamp). The more waypoints are changed during that interval, the more the precision increases for that alert.

The *Monitoring Precision* (P_M) is conceptually opposite to P_A , due to it penalizes the waypoint interactions performed during *monitoring time*, so the less interactions here, the more precision obtained. It is computed as

$$P_M(s) = \frac{1}{1 + |W_M(s)|} \quad W_M(s) = \overline{\bigcup_{i \in A(s)} W_a(s)},$$

where $W_M(s)$ is the set of all waypoint interactions performed during monitoring time, i.e, the complementary of all waypoint interactions made to avoid incidents.

4.1.7. Reflexes

The *Reflexes* (*R*) metric is devoted to rate the fastness with which an operator responds when an alert is triggered during the mission. Given a Maximum Response Time (MRT) (fixed to 15 seconds in this work), and the set of alerts of a simulation $A(s)$, we define, for every $a \in A(s)$, the set $I_a(s)$ comprising all the interactions performed during the response time interval for that alert, i.e:

$$I_a(s) = \{i \in I(s) \mid t(a) \leq t(i) \leq (t(a) + MRT)\},$$

where $t(a)$ and $t(i)$ are the timestamps of the alert and the interactions respectively. For every interaction belonging to $I_a(s)$, a reflexes rating (rr) is assigned as follows:

$$rr(i, a) = 1 - \frac{t(i) - t(a)}{MRT}$$

This rating decreases from 1 to 0 linearly as the timestamp of the interactions move away from the alert trigger time. Finally, averaging these ratings over all the alerts in the simulation, the equation of the Reflexes metric is obtained:

$$R(s) = \frac{1}{|A(s)|} \sum_{a \in A(s)} \frac{1}{I_a(s)} \sum_{i \in I_a(s)} rr(i, a) \quad (6)$$

As it can be seen, this metric is similar to the precision metric described above, in the sense that both of them are averaged over the set of mission alerts $A(s)$. However, while precision takes into account the whole set of mission interactions, reflexes do not care about the interactions made during out-of-alerts periods.

4.2. Obtaining a robust discrimination of performance metrics

Computing the 7 metrics defined above for all the simulations dataset results in a 7-dimensional metric space, on which we can apply **clustering** methods to group together simulations which have similar performance profiles. The problem here consists of finding an optimal discrimination of the metric space, for which we must select carefully both the number of clusters and the clustering algorithm (See Figure 2).

Usually, experts in the field of UAVs, specially those devoted to train UAV operators, ask for processes of behavioral discrimination as the one performed here, and the main requisite asked is: “Finding a set of behavioral profiles, such that operators sharing one profile are very close to each other, and those not sharing profiles present significantly different behaviors” [8].

In terms of a clustering result, this requisite is reflected by the combination of two properties: The *compactness*, which assess cluster homogeneity, usually by looking at the intra-cluster variance, and the *separation*, which quantifies the degree of separation between clusters. The combination of these two opposing trends has been named as “Quality" of a clusterization in this work. For the quality validation of the clusters, we selected three internal validation measures from the state of the art that combine compactness and separation into a single score: *Dunn Index (DI), Silhouette Width (SW) and Calinski-and-Harabasz index (CH)* (see Section 2.3).

Apart from the quality of the discrimination, which would suppose the main functional requisite of this process, it is also worth taking into account the *stability* of the clustering solution as a non-functional requisite. For this work, two stability metrics have been selected for their simplicity and good results: *Average proportion of non-overlap (APN)* and *Average distance (AD)* (see Section 2.3).

The validation process carried out in this work consists simply of computing clustering solutions and applying the validation measures described above, looping over different selections of two main parameters: the clustering algorithm and the number of clusters to find (See Figure 2).

Let α be the set of clustering algorithms used, and ν the set of possibilities for the number of clusters. After the validation loop, we get a total of 5 *validation lists* (one for each validation measure) of length $\alpha \times \nu$. Each element of a validation list is an identifier of the combination *algorithm-number of clusters*

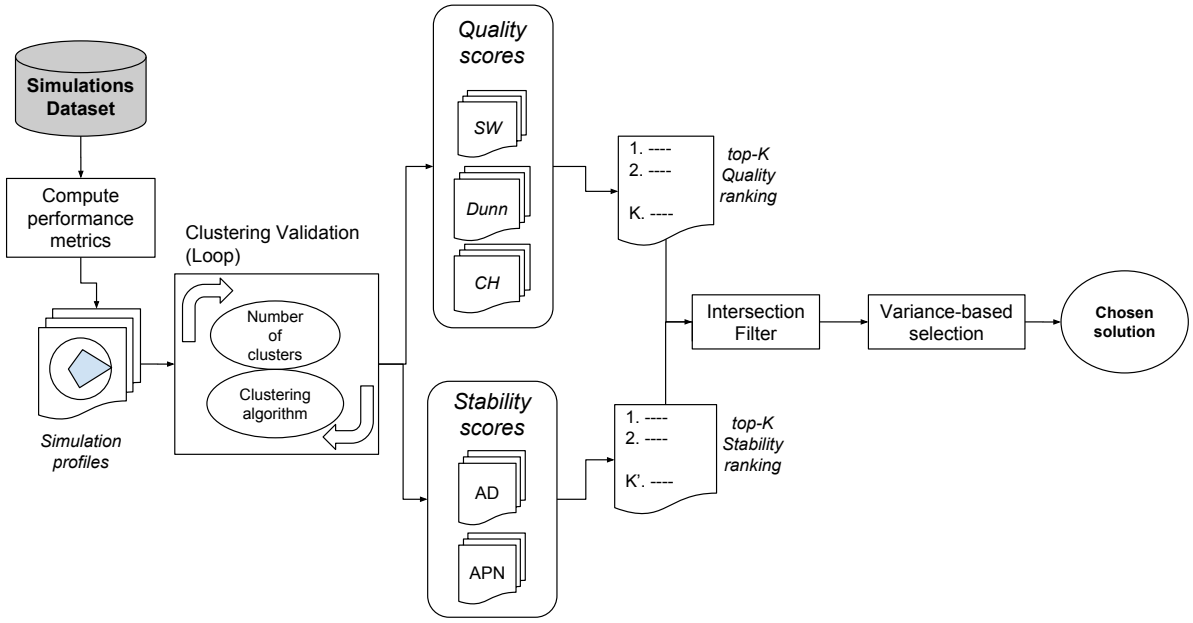

Fig. 2. General scheme of the clustering validation and selection process carried out in this work.

(e.g: “kmeans-6"). These lists are ordered, based on the corresponding score lists and divided into two groups: *Quality rankings*, for the three quality validation measures mentioned above, and *Stability rankings*, for the two stability measures. For each of the validation rankings, we select the best top-K results which passes to the selection process (See Figure 2, top-K rankings).

Since we aim to achieve results that represent a fair balance between quality and stability, we must filter the rankings preserving only those elements appearing in both of them (See Figure 2, Intersection Filter). By applying this filter, we ensure that all the possible chosen solutions offer a compromise between quality and stability. If case the intersection filter results empty, we will increase the value of K, to extract a bigger amount of solutions from the validation lists.

Finally, Once applied this filter, the criteria for choosing a single clustering solution is computing the variance of the scores for each validation measure, and getting the solution winning the ranking of the most variable validation measure (See Figure 2, Variance-based selection). With this, we are choosing the most representative winner of all rankings.

5. Experimental Setup

The experiments have been focused on answering the following research questions:

- **RQ1 Are the performance metrics suitable in terms of redundancy and information provided?** This question is studied in section 6.1.
- **RQ2 What clustering solution provides the best discrimination guaranteeing a fair balance between quality and stability?** This question is analyzed in section 6.2.
- **RQ3 What type of performance profiles are extracted from the best clustering results?** This question is answered in section 6.3.

5.1. Dataset

In this experiment, the simulation environment (DWR) was tested with Computer Engineering students of the Autonomous University of Madrid (AUM), all of them inexperienced in HSC systems. Although the experiment was conducted in many different days, all users received the same tutorial before using the simulator, so, thus no distinction is made between the experiment days, and all data extracted during these days is therefore treated uniformly.

The dataset resulted of extracting data from this experiments is composed of 361 distinct simulations, played by a total of 60 users. In order to achieve a robust analysis of the data extracted, we must clean the dataset by removing those simulations which can be considered as useless. Two different criteria have been

Table 1
Parameter tuning for all the variables involved in the experiments of this work.

Context	Parameter	Value
DWR	Maximum Response Time (MRT)	15 s
Metric analysis	Correlation cutoff	0.75
Clustering	Clustering Algorithms	K-means SOM SOTA PAM AGNES
	Minimum number of clusters	3
	Maximum number of clusters	8
	Metric to determine distances	Euclidean
	Agglomeration method (for AGNES)	average link
	K - Top ranking size	5

applied to identify useless simulations: 1. A simulation should last at least 20 seconds. 2. A valid simulation should include, at least, the triggering of one alert during the course of the mission. From the 361 simulations composing the initial dataset, only 149 of them are considered useful simulations, and will be used in the data analysis process.

5.2. Parameter Tuning

The choice of the parameters involved in the process described in the previous section is very important for the success of the behavioral analysis. Table 1 gathers the tuning of all the necessary parameters for the whole process. Among them, the most relevant for this work are those related to *Clustering*. We make use of five clustering algorithms: *K-means, SOM, SOTA, PAM and AGNES* (See Section 2). The number of behaviors (clusters) to find will go from 3 to 8 (6 possibilities), due to it is considered that more than 8 behaviors would produce an over-fitted intractable behavioral model. A total of 30 (5×6) possible clustering solutions will be computed in the validation loop.

6. Experiments

The purpose of the experiment carried out using the simulation environment DWR is to analyze the behavior and performance of inexperienced UAV operators during a training session. Once we have a robust dataset, we will assess the performance of every user following the metrics defined above, and based on this evaluation, we will create and group users in order to create profiles that indicate similar user behaviors. Finally, those clusters will be analyzed and interpreted in the context of this experiment.

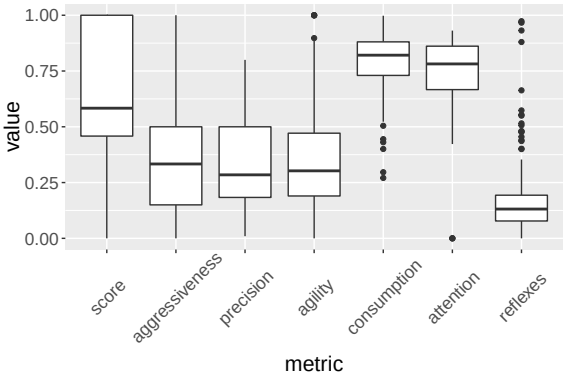

Fig. 3. Distribution of the proposed performance metrics for the simulation environment DWR, within the dataset used in this work.

In order to have a better understanding of the different key aspects involved in the whole experiment, we will progressively answer to each of the research questions formulated in Section 5.

6.1. Metrics Information and Correlation

After computing the set of performance metrics for all the simulations in the dataset, it is worth examining their distributions and extract general information about the performance of the users in the experiment.

Figure 3 shows the distribution box plots for each of the performance metrics proposed in 4.1. Among them, *Consumption*, *Attention* and *Reflexes* are the ones with less variance, and thus, they can be used to extract some general information of the type of data used in the analysis.

- High levels of consumption may indicate that users tend to play short-time simulations, probably aborting them even before accomplishing the mission objectives.
- High levels of attention may represent interaction overloads and restless behaviors, in which the operator, even after being advised of the importance of maintaining calm in this type of environments, tend to try the different commands available in the simulator unconsciously.

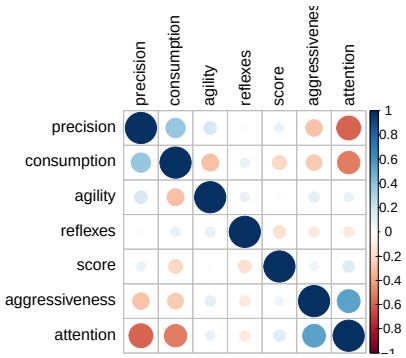

Fig. 4. Correlation Matrix of the performance metrics proposed for the simulation environment DWR, within the dataset used in this work.

– Low levels of reflexes are a clear sign of the lack of experience of the users from these experiments. Since they scarcely know how to execute the different interactions available in the simulator, it is quite difficult for them to respond quickly to an incident.

All these factors, and to a lesser extent the distribution of the rest of the metrics, reinforce the main notion of this dataset, which is the *inexperience* proper from young students. Even though, not all the metrics show a cumulative distribution, thus it is necessary to deepen the behavioral analysis with the clustering techniques proposed in this work.

With regard to the correlation of the metrics and the possible redundancy among them, we analyze the correlation matrix shown in Figure 4 in order to get insight about these topics. From this matrix, we check that none of the correlations found surpass the correlation threshold established for this experimentation (0.75, as mentioned in Section 5), and thus, we can ensure the non-redundancy of the whole set of metrics, and justify its use for the subsequent experiments.

6.2. Clustering Validation

The main issue of this experiment consists in applying the validation process described in Section 4.2 in order to obtain an optimal discrimination of groups of performance in a simulations dataset from the simulator DWR, which will allow an expert in the system to develop a robust behavioral analysis of the users summarizing the dataset.

The first step of the process applies a set of clustering validation measures to assess both: the quality and the stability of a clustering solution. A total of 8 algorithms have been tested, over 6 different possibilities for the number of clusters, hence there are 48 possible clustering solutions for this experiment (See Table 1). After that, we select the 5 best clustering solutions for each of the validation measures. The top-5 ranking from this process is shown in Table 2, along with the associated validation scores for each solution.

From this table, we can have a general overview of the algorithms and number of clusters that work better for either the quality and the stability. With regard to the clustering algorithms, we find that K-means and AGNES dominate the quality subtable with a 40% and 33% of appearance respectively, although AGNES is the absolute winner for the Dunn Index. In this case, the solution chosen by the silhouette is based on centroids, hence K-means have identified specific positions where it can generate a strong discrimination. For AGNES the solution is related to the smallest number of clusters, which usually obtains the highest values when no patterns are found. Looking at the stability subtable we check that SOTA is the most common algorithm, along with PAM with a 30% of appearance for both of them. Thus, it is clear to see that quality and stability are somehow complementary, and finding an algorithm optimizing both of them is not trivial.

Regarding the number of clusters, values 3 and 5 dominate for the quality measures, with a percentage of appearance around 25% for each of them. In contrast, for the stability we find values from 6 to 8 more commonly. This fact is somewhat surprisingly, due to high number of clusters usually lead to present small clusters, which tend to be less stable. However, it may be the case that the size of the clusters is more balanced only when the number of clusters is high, due to the fine granularity of the underlying data distribution.

As it was said in section 4, our criteria to select a clustering solution from this set of rankings is to choose the winner from metric with highest variance, providing that the solution appears in both the quality and the stability rankings. In this experiment, before applying this technique, we normalize each of the validation score lists, using a common unity-based normalization, putting all of them into the interval $[0, 1]$. With this, the importance (variance) for each of the validation measures results as follows:

– *Quality ranks*:
$[DI = 0.048, \mathbf{SW{=}0.062}, CH = 0.051]$
– *Stability ranks*: $[APN = 0.053, AD = 0.05.]$

Table 2

Top-5 (of 30) ranking for each of the clustering validation measures used in this work, for both the quality and the stability aspects of a clustering solution. Each cell contains an identifier of the clustering solution parameters (algorithm-number of clusters) and the associated validation score between brackets. Cells in italic represent the winner of each ranking, providing that it appears in both quality and stability lists. Bolded cell represents the chosen solution for analysis.

	Quality measures			**Stability measures**	
#	**DI**	**SW**	**CH**	**APN**	**AD**
1	*AGNES-3 (0.21)*	**KMEANS-7 (0.251)**	KMEANS-3 (43.077)	*AGNES-3* (0.077)	*KMEANS-8 (0.53)*
2	AGNES-4 (0.205)	KMEANS-6 (0.248)	DIANA-3 (40.956)	SOTA-3 (0.137)	SOM-8 (0.53)
3	AGNES-7 (0.182)	KMEANS-8 (0.242)	SOM-3 (40.737)	AGNES-4 (0.137)	KMEANS-7 (0.543)
4	AGNES-8 (0.182)	SOM-6 (0.238)	KMEANS-4 (40.186)	AGNES-5 (0.163)	SOTA-8 (0.546)
5	AGNES-5 (0.148)	KMEANS-4 (0.236)	SOM-4 (40.164)	SOTA-4 (0.184)	PAM-8 (0.548)

Thus, the selected solution is KMEANS-7, since it is the winner of the SW rank, and it also appears in one of the stability ranks (AD, position 5).

6.3. Final Profiles

Once we have validated and selected the optimal clustering solution for our experimental dataset, we must analyze the clusters obtained in order to interpret them, and to define which behavioral patterns are found among the operators participating in the experiment. This is usually done by an expert in the domain and the simulation environment, but here, since we know well the mechanics of the simulator [2], we can also perform a simple cluster analysis.

According to the results obtained in this experiment, K-means with 7 clusters turns out to be the optimal clustering solution for this dataset. Figure 5 shows the centroids (most representative elements) for each of the clusters, in the form of radial plots under the performance metrics space.

Below are detailed the explanation and behavioral patterns extracted from each of the clusters obtained by applying the KMEANS-7 clustering solution:

- *Cluster 1*: This profile gathers operators with high precision but low agility and and attention rates, which is and indication of *awareness* about the mission incidents.
- *Cluster 2*: This cluster represent a strange balance between high reflexes and and low precision values, which means that operators interacted fast against incidents, but not for the purpose of solving them. This demonstrates that these users are completely unaware of the mission objectives, which is also shown is their extremely bad scores.
- *Cluster 3*: In this cluster, operators achieve good score rates with high agility and low precision values, which is a sign of a *restless*, but *high-skilled* behavior. It is the biggest cluster of all (with a size of 37.21%), and thus the best representation of the type of users of this dataset.
- *Cluster 4*: This cluster is small and remarkable for its huge levels of reflexes in comparison with the rest of the clusters. Since its aggressiveness is low, operators probably tried to solve the incidents quickly by moving waypoints, instead of creating new paths. As it can be seen by the score metrics, that did not lead to good results.
- *Cluster 5*: In this cluster, the high values of Agility, Precision and Reflexes make us think that this type of operators were aware about the incidents of the mission, and tried to fight them quickly. However, they run the simulation so fast that could not achieve good scores. Probably, these students were somehow *bored* in the experiment and they tried to complete the simulation as fast as possible.
- *Cluster 6*: These are the best users of all in terms of the scores obtained in the simulations. They performed soft interactions (low aggressiveness) as moving waypoints, to efficiently react against the incidents quickly, and did not change the simulation course when it was not needed. Undoubtedly, this cluster represents the most desired behavior for a expert operator.
- *Cluster 7*: This profile gathers operators featuring extremely high levels of aggressiveness. Probably, they were *unfocused*, and spent the whole simulation time creating new paths for all the UAVs in the mission.

[2] The simulation environment DWR was developed and tested by our research group (See [30])

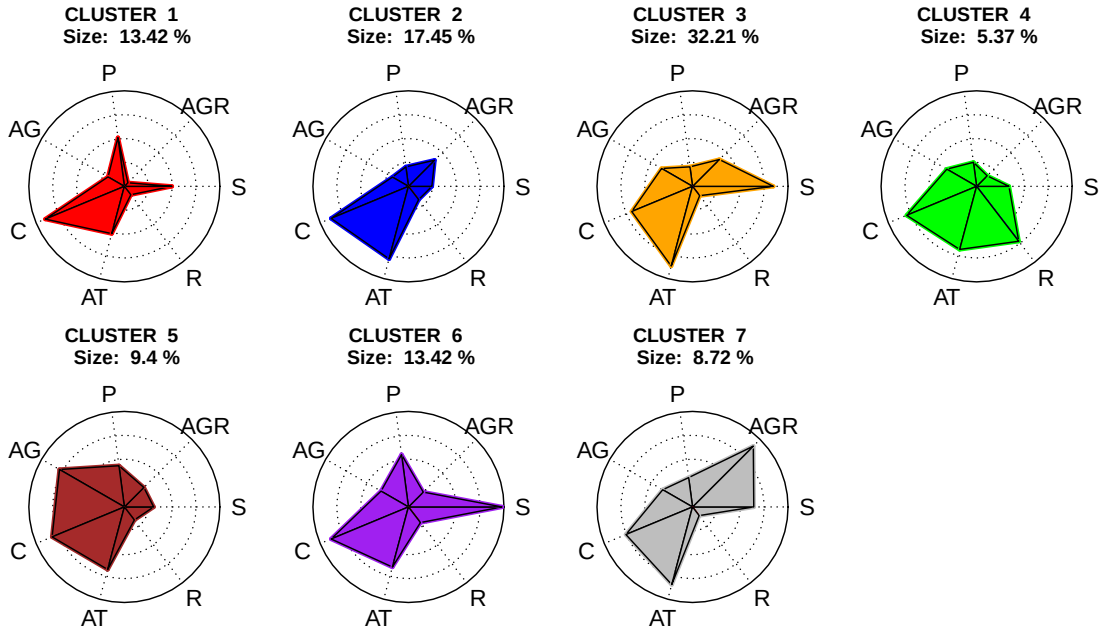

Fig. 5. Centroids obtained by applying KMEANS with 7 number of clusters in the experimental dataset.

Also, this profile analysis can support the decision making for a UAV instructor, when he/she needs to decide which users pass a training session and can opt to be candidates for future training procedures. In our experimental validation, the best profiles are found for Cluster 7, so that would be the selected group of potential UAVs operators.

7. Conclusions and Future Work

In this work, a multi-UAV simulation environment has been used to carry out an experiment with inexperienced operators, in order to discover behavioral patterns from their performance during the simulations.

To achieve this, three main steps have been followed: First, a set of seven performance metrics have been designed in order to define the operator performance profile, measuring the quality of his/her interactions during a simulation. Then, the performance metrics are introduced into several *clustering algorithms*, to discover some groups or patterns in the operator performance. In order to obtain an optimal clustering solution, we developed an intensive validation process, in which we rank a battery of possible solutions in terms of several measures, and then decide what solution obtains the best balance between quality and stability.

The experimental results show that both the performance metrics created and the chosen clustering result characterize well the behavior of the users of the experiment, which proves the validity of the methodology for this simulation environment.

As future work, it is intended to extend the experimentation made in this work by adding bigger datasets, which would lead to a more robust and rich cluster analysis. Also, an abstraction of this whole process, and its application and comparison for multiple interactive environments would be desirable for the community of performance analysis. This entails the use of general performance metrics for any Human-Computer Interaction system, and the definition of ground-truth performance profiles, for easily comparing the expected behaviors with those obtained by the methods developed in this work.

Acknowledgments

This work has been supported by the next research projects: EphemeCH (TIN2014-56494-C4-4-P) Spanish Ministry of Economy and Competitivity, CIBERDINE S2013/ICE-3095, SeMaMatch EP/K032623/1, all of them under the European Regional Development Fund FEDER, and Airbus Defence & Space (FUAM-076914 and FUAM-076915). The authors would like to acknowledge the support obtained from Airbus Defence & Space, specially from Savier Open Innovation

project members: José Insenser, Gemma Blasco, Juan Antonio Henríquez and César Castro.

References

[1] Arabie, P., Carroll, J.D., DeSarbo, W., Wind, J.: Overlapping clustering: A new method for product positioning. Journal of Marketing Research pp. 310–317 (1981)

[2] Ayaz, H., Shewokis, P.A., Bunce, S., Izzetoglu, K., Willems, B., Onaral, B.: Optical brain monitoring for operator training and mental workload assessment. Neuroimage 59(1), 36–47 (2012)

[3] Bayer, U., Comparetti, P.M., Hlauschek, C., Kruegel, C., Kirda, E.: Scalable, behavior-based malware clustering. In: NDSS. vol. 9, pp. 8–11. Citeseer (2009)

[4] Beard, B.R.W., McLain, T.W., Nelson, D.B., Kingston, D., Johanson, D.: Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. Proceedings of the IEEE 94(7), 1306–1324 (2006)

[5] Beard, R.W., McLain, T.W.: Multiple uav cooperative search under collision avoidance and limited range communication constraints. In: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on. vol. 1, pp. 25–30. IEEE (2003)

[6] Begis, G.: Adaptive gaming behavior based on player profiling (Aug 22 2000), uS Patent 6,106,395

[7] Berry, M.W., Castellanos, M.: Survey of text mining. Computing Reviews 45(9), 548 (2004)

[8] Boussemart, Y.: Predictive models of procedural human supervisory control behavior. Tech. rep., DTIC Document (2011)

[9] Boussemart, Y., Cummings, M.L., Fargeas, J.L., Roy, N.: Supervised vs. unsupervised learning for operator state modeling in unmanned vehicle settings. Journal of Aerospace Computing, Information, and Communication 8(3), 71–85 (2011)

[10] Caballero, F., Merino, L., Ferruz, J., Ollero, A.: A visual odometer without 3d reconstruction for aerial vehicles. applications to building inspection. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. pp. 4673–4678. IEEE (2005)

[11] Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods 3(1), 1–27 (1974)

[12] Chen, Y.M., Dong, L., Oh, J.S.: Real-time video relay for uav traffic surveillance systems through available communication networks. In: Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE. pp. 2608–2612. IEEE (2007)

[13] Datta, S., Datta, S.: Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics 19(4), 459–466 (2003)

[14] Drury, J.L., Scholtz, J., Yanco, H.A.: Awareness in human-robot interactions. In: Systems, Man and Cybernetics, 2003. IEEE International Conference on. vol. 1, pp. 912–918. IEEE (2003)

[15] DunnâĂă, J.C.: Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics 4(1), 95–104 (1974)

[16] Goodrich, M.A., Morse, B.S., Gerhardt, D., Cooper, J.L., Quigley, M., Adams, J.A., Humphrey, C.: Supporting wilderness search and rescue using a camera-equipped mini uav. Journal of Field Robotics 25(1-2), 89–110 (2008)

[17] Herrero, J., Valencia, A., Dopazo, J.: A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17(2), 126–136 (2001)

[18] Honkavaara, E., Saari, H., Kaivosoja, J., Pölönen, I., Hakala, T., Litkey, P., Mäkynen, J., Pesonen, L.: Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight uav spectral camera for precision agriculture. Remote Sensing 5(10), 5006–5039 (2013)

[19] Jiang, S., Wang, L.X.: A clustering-based feature selection via feature separability. Journal of Intelligent & Fuzzy Systems (Preprint), 1–11

[20] Kaufman, L.R., Rousseeuw, P.: Finding groups in data: An introduction to cluster analysis. Hoboken NJ John Wiley & Sons Inc (1990)

[21] Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological cybernetics 43(1), 59–69 (1982)

[22] Larose, D.T.: Discovering knowledge in data: an introduction to data mining. John Wiley & Sons (2014)

[23] MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA. (1967)

[24] McCarley, J.S., Wickens, C.D.: Human factors concerns in uav flight. Journal of Aviation Human Factors (2004)

[25] McCarley, J.S., Wickens, C.D.: Human factors implications of UAVs in the national airspace. University of Illinois at Urbana-Champaign, Aviation Human Factors Division (2005)

[26] McKinley, R.A., McIntire, L.K., Funke, M.A.: Operator selection for unmanned aerial systems: comparing video game players and pilots. Aviation, space, and environmental medicine 82(6), 635–642 (2011)

[27] Nikolos, I.K., Valavanis, K.P., Tsourveloudis, N.C., Kostaras, A.N.: Evolutionary algorithm based offline/online path planner for uav navigation. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 33(6), 898–912 (2003)

[28] Pereira, E., Bencatel, R., Correia, J., Félix, L., Gonçalves, G., Morgado, J., Sousa, J.: Unmanned air vehicles for coastal and environmental research. Journal of Coastal Research pp. 1557–1561 (2009)

[29] Ramírez-Atencia, C., Bello-Orgaz, G., R-Moreno, M.D., Camacho, D.: Branching to find feasible solutions in unmanned air vehicle mission planning. In: Intelligent Data Engineering and Automated Learning–IDEAL 2014, pp. 286–294. Springer (2014)

[30] Rodriguez-Fernandez, V., Menendez, H.D., Camacho, D.: Design and development of a lightweight multi-uav simulator. In: Cybernetics (CYBCONF), 2015 IEEE 2nd International Conference on. pp. 255–260. IEEE (2015)

[31] Rodríguez-Fernández, V., Menéndez, H.D., Camacho, D.: User profile analysis for uav operators in a simulation environment. In: Computational Collective Intelligence, pp. 338–347. Springer (2015)

[32] Rodríguez-Fernández, V., Menéndez, H.D., Camacho, D.: Automatic profile generation for uav operators using a simulation-based training environment. Progress in Artificial Intelligence 5(1), 37–46 (2016)

[33] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20, 53–65 (1987)