Grant Agreement No. 611373

FP7-ICT-2013-10

D3.2 - Symbolic language interpreter

Due date of deliverable: 31/01/2016
Actual submission date: 04/02/2016

Start date of project: 01 January 2014 Duration: 36 months

Organization name of lead contractor for this deliverable: CNR

Revision: version 1

Dissemination level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

1 Contents

2 Outline of the deliverable 2

3 Gesture detection and classification 3

3.1 Hand detection 3

3.2 Gesture classification 4

4 Gesture sequence interpretation 7

4.1 Syntax checker 7

5 Association to mission actions 10

6 Validation trials 12

6.1 Experiment description 12

6.2 Result statistics 15

7 Conclusions 18

8 Annexes 19

8.1 Annex A - Code gestures table 19

8.2 Annex B – Commands, Sequence, semantics table 21

2 Outline of the deliverable

This deliverable describes the design, implementation and validation methodologies of the software modules needed to achieve the task of symbolic language interpretation.
The deliverable reports the chain of operations requested in order to: i) recognize the sequence of gesture performed by the diver; ii) analyze the gesture sequence, validate the correctness and produce a set of mission commands; iii) execute the proper mission task to fulfil the gesture sequence request.
The validation of the operation chain is validated through the analysis of experimental results carried out during the field trials in Biograd Na Moru (Croatia), October 2015, and Genova (Italy), November 2015.

3 Gesture detection and classification

In this section we describe the methods used to recognition the gestures performed by the divers. It is important to mention that all divers received an introduction to the CADDIAN language presented in Deliverable 3.1. We can divide this section into two tasks: hand detection and gesture classification. Figure 1 shows a general block diagram of the classification system. First we receive as input the camera images; these are processed parallel by two methods, one that computes the 3D information and the other that analyses only the monocular images. The first one creates a disparity map to detect the hands by thresholding the distance to the camera, and the second one passes the image to a Haar cascade classifier. Afterwards, the output of these techniques are compared to see if there are any inconsistencies, specifically on the disparity map. These processes describe the hand detection task; then the information and 2D patch covering the hands is introduced into an existing classification method to output a result.

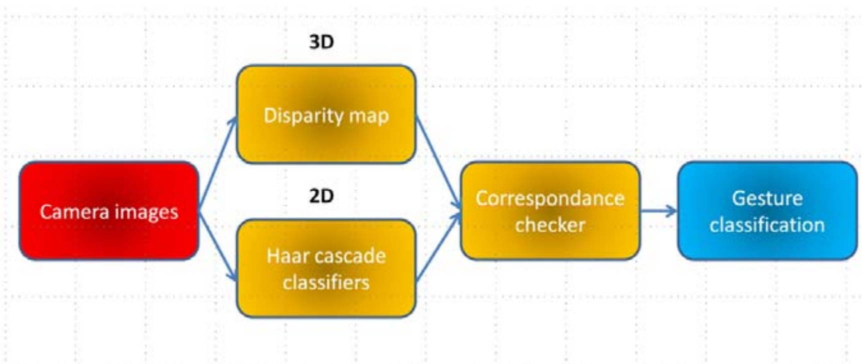


Figure 1. General block diagram for gesture classification. The yellow blocks are part of the hand detection task and the blue one to the classification task.

3.1 Hand detection

Since the diver is in constant motion underwater it is difficult to establish precise constraints for the hand's pose; as a matter of fact, the diver's body pose changes greatly. It can be in horizontal or vertical position, and not directly facing to the camera. For this reason, a robust method to locate the hands on the camera images is needed.

The first implemented method creates a disparity map from stereo images, and then under the assumption that the hands are the closest "objects" to the camera, filters out the rest of the image. If a hand is successfully detected, further information like pose and finger location can be computed. This is shown on the right side of Figure 2 where the fingers are colored in red, the palm in in blue and the wrist in green; also the pose of the wrist is shown with an arrow.

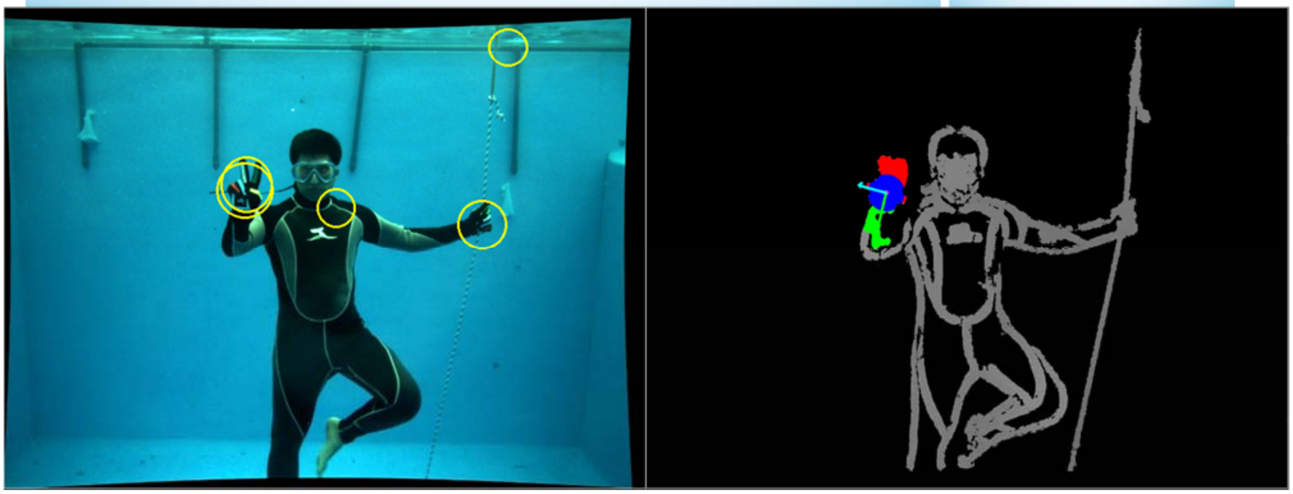


Figure 2. (Left) Monocular image with the hand candidate patches output by the Haar classifier. (Right) Disparity image with the detected hand segmented

However, the position of the diver is not always stable. For example, in bad weather conditions and close to the surface the divers are continuously moved around by the waves. This is more likely to cause inaccurate disparity maps due to errors in the feature matching process from the stereo images. These errors are commonly present in underwater scenarios because there are scarce image features, for example the background and the diver suit are uniformly colored. For this reason, the system cannot always rely on the disparity map to locate the hands. Figure 2 shows, how the disparity information is only available where there are edges e.g. the transition between the diver's body and the background.

To make the system more robust a Haar cascade classifier was trained on monocular images from the hands. This classifier was chosen because it can scan the whole image for possible hand candidates at different scales at a higher speed than the camera's frame rate. In this way, even if the diver moves closer or further away from the camera, the classifier will be able to detect the hands.

Nonetheless, usually Haar classifiers need several thousand images to have high precision and there are no available datasets for underwater imagery because they are difficult to obtain. Thus, the Haar cascades were trained to underfit the data (high recall) and be able to detect hands in every frame at the cost of having a high number of false positives. It is then in the classification step where these false positives are filtered out and a classification for the hand gesture is given. On the left side of Figure 2 the image patches that are the candidate locations for the hands are shown.

3.2 Gesture classification

Once the possible hand candidates are detected, these image patches are encoded into feature vectors and used as input for the final classifier which is a proposed variation of a Random Forest: Multi-Descriptor Nearest Class Mean Random Forest (MD-NCMF). This method was proposed because it can aggregate different images descriptors without loss of information, which is extremely useful because different descriptors are invariant to different type image distortions i.e. some are more invariant to illumination changes, others to change in viewpoint, etc. Underwater imagery presents several type of distortions due to light backscatter in water, which can be present in some of the video frames or nor; for this reason, a classifier that can cope with most of these image phenomena is necessary. The details of this classifier are explained in the paper published in OCEANS Genova 2015: Visual diver detection using MD-NCMF in the context of underwater Human Robot Interaction.

After some experimentation, the feature descriptors Histogram of Oriented Gradients (HOG) and HSV color representation showed better results. Overall, 16 different type of gestures are recognized; which are then the input of the CADDIAN syntax checker explained in the next sections, and which is able to recognize complex messages. This classification pipeline has proved to be robust against scale (different distance of the diver to the camera) and continuous movement of the diver to harsh weather conditions. Figure 3 shows different detected and classified gestures. In section 5, more examples of gestures in different water conditions are shown and explained.

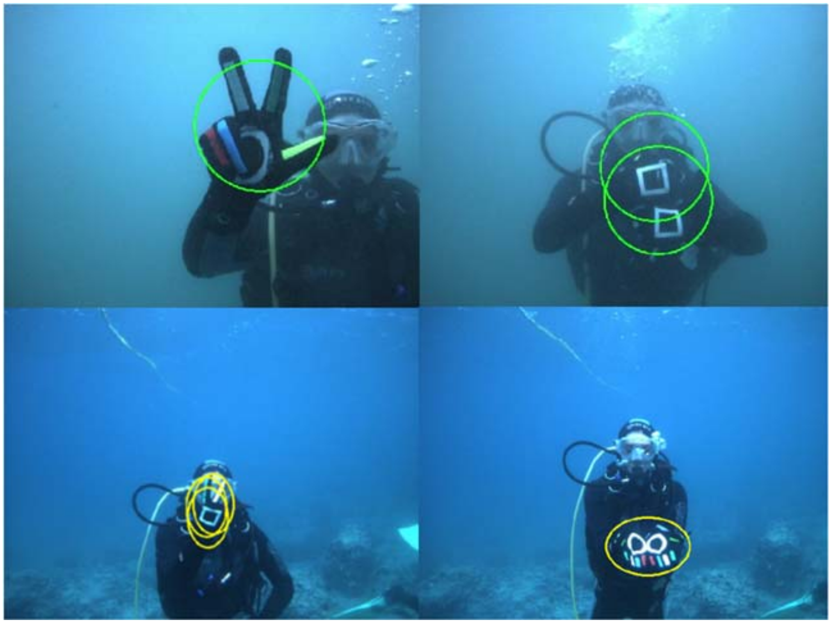


Figure 3. Detected gestures: take a photo, carry equipment, start communication, go to the boat. Starting from the top left in clockwise fashion.

Finally in Figure 4, a screenshot of the system output is displayed; the system detects each gesture performed by the diver if it is stable through several frames; this means that the same gesture is detected continuously, in this way sporadic false positives are avoided. The system also keeps track of the last recognized gesture.

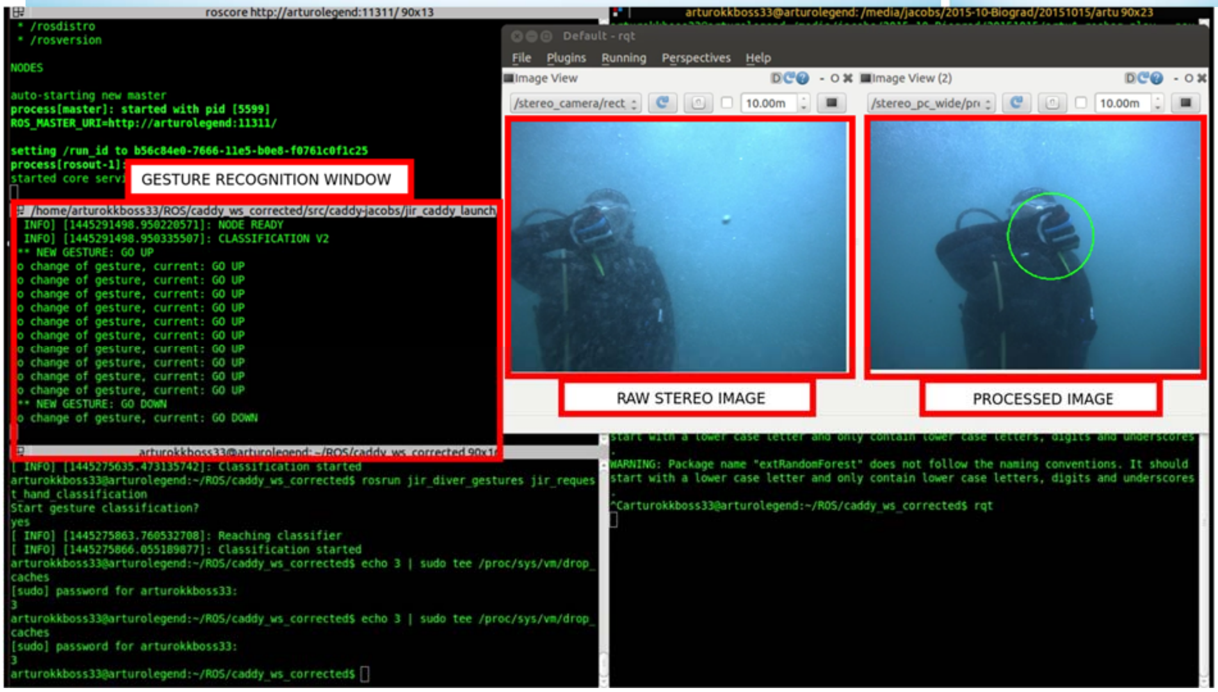
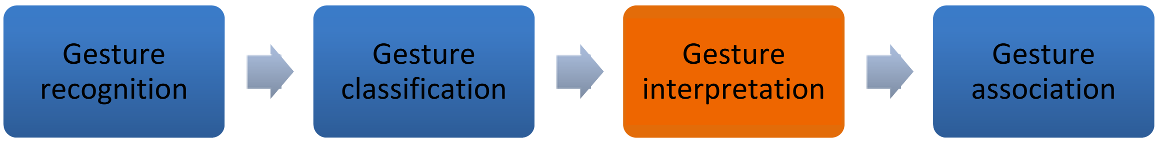


Figure 4. Classification system processing the raw stereo images, processing them to detect the hands and classify them. The output is shown in the gesture recognition window

4 Gesture sequence interpretation



The gesture sequence interpretation is implemented through a parser that accepts syntactically correct sequences and reject the wrong ones. The parser accepts commands or messages which are, to all intents and purposes, sequences of gestures. Each gesture, as described in “Deliverable 3.1”, can be represented by a symbol. In this way a message/command is a sequence of symbols/gestures delimited, at the beginning, by a symbol of “Start communication” and, at the end, by the same symbol or by a symbol of “End of communication” (see Figure 5). According to which symbol has been found after the initial one (i.e. “Start communication”), the Syntax Checker goes on by applying the Caddian syntax rules.

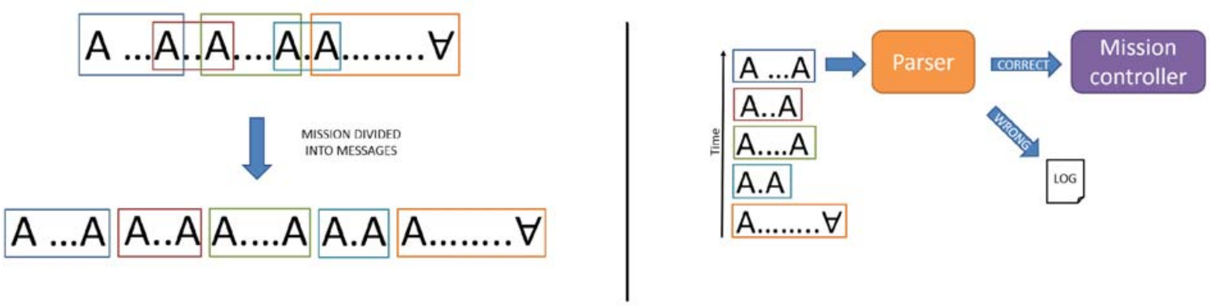


Figure 5. On the LEFT Mission segmentation into single messages; on the RIGHT Single messages processed by the parser

Missions (a sequence of commands/messages, see “Deliverable 3.1”) are segmented into messages and the syntax checker (parser) parses each message. Then, the parser can accept the sequence that is to all effect, because of the acceptance, a command and the command is passed to the mission controller. In case the sequence is wrong, the error is logged and a warning is issued to the diver.

The Syntax Checker, better detailed in the next section, has been realized as a ROS node that checks and validates each command before allowing the robot to execute it. Gestures Classifier and Gestures Interpreter communicate through a predefined standard translation table, which translates gestures to integer numbers as can be seen in Annex A 8.1.

Gestures Interpreter, in turn, communicates with Mission Controller following a predefined coding, which can be seen in Annex B 8.2.

4.1 Syntax checker

Each sequence performed by the diver in the CADDIAN language must undergo a syntax check to be validated before being passed to the robot for the require task execution. Such a check is based on the syntactic rules of CADDIAN that are applied to the sequence to understand if it has a correct structure (only its syntax is checked but not its semantics).

During these validation trials only a subset of (static) gestures has been taken into account. These gestures could be subdivided into the following categories: Number, Caddian-related, Slang, Emergency, Direction, Work and Place.

These groups are structured as follows: a) Number includes numbers from 1 to 5; b) Caddian-related consists of open and close communication and number delimiter; c) Slang includes boat, out of breath, out of air, problem and danger; d) Emergency is made up of general evacuation; e) Direction groups go up, go down, go forward and go backward; f) Work consists of mosaic, photo and carry; g) Place includes boat and here.

The considered gestures belonging to the Caddian-related group are useful for sequence segmentation: the diver can issue many different complex commands in a row before concluding with the close communication one. Each complex command must start with the open communication symbol, to allow its segmentation; each complex command can be made up of one or many gestures.

According to which gesture has been found after the open communication, the Syntax Checker goes on by applying the Caddian syntax rules; in particular:

1. if the second symbol belongs to the Slang category and the following one is either an open or a close communication, the Syntax Checker validates the sequence, since the slang gestures are performed alone (apart for the Caddian-related symbols) in a “quick” communication fashion;
2. if the second symbol belongs to the Direction group, the Syntax Checker verifies the following symbols: if the subsequent gestures are (one or even more) digits of a number, followed by its number delimiter and then by an open or close communication, then the sequence is marked as valid;
3. if the second symbol belongs to the Work group, the Syntax Checker distinguishes between works that require one or more arguments and works without arguments. In the first case, the possible arguments (i.e. the subsequent gestures in the sequence) could be either a Place or one or two Numbers (each one consisting of one or more digits) plus their number delimiters. After that, if an open or a close communication is found, the sequence is validated. In the second case, the sequence is validated only if the subsequent (i.e. third) gesture is either an open or a close communication;
4. if the second symbol belongs to the Emergency category, similarly to the Slang group, the Syntax Checker validates the sequence only if the successive symbol is either an open or close communication;
5. if after the first open communication symbol, the Syntax Checker finds either another open communication or a close communication gesture, the sequence is marked as valid, corresponding to the null command (no operations);
6. in all the other cases the Syntax Checker marks the sequence as non valid (e.g. if the second symbol belongs to the Number or Place group).

For the first validation trials, the set of recognized gestures has been restricted but it will be integrated with all the other ones in a future stage and the Syntax Checker will be extended as well to validate sequence containing also these additional symbols. Figure 6 shows a screenshot of the classification system, on the left hand side a list (history) of the detected gestures can be seen and on the right side the monocular image with the detected gesture. If the list of gestures obey the rules previously described i.e. it is approved by the syntax checker, then and only then the command code is sent to the UAV. In this case, a “Start communication” signal was received, then “Up” a Direction symbol, a “Number”, “Number Delimiter” and “End communication.”

Figure 6. Classification system detecting hands and labelling individual gestures. If a "Start" and "End" communication gestures is received, the in-between symbols are checked by the syntax checker and sent to the AUV if correct.

5 Association to mission actions

The aim of providing a compliant behavior of the overall robotic system, with respect to the command issued by the diver, is made available through the development of an automatic selection system for the execution of the proper autonomous robotic tasks.
First of all, the basic CADDY functionalities has to be mapped into subsets of tasks that can be provided by the robotic platforms. In order to define the primitives-tasks matching, an additional high-level task set has to be defined as cross-interface between the primitives and robotic task sets.

A preliminary definition of the three sets is reported in Figure 7, where:

- *functional primitives* represent the macro-actions that the robotic platform has to carry out in order to support the diver operation and that are strictly related to the current functional mode (slave, guide, observer). The primitives are triggered by the recognized gestures;
- *high-level logical tasks* are the interface between the primitives and the operative task provided by robot. This logical task set is common in the overall architecture and will provide the required functionalities activating the proper low-level tasks that are currently made available by the employed robotic platform;
- *low-level robotic tasks* are the actual implemented autonomous functionalities on the target robot, e.g. speed regulators, heading and depth controller, etc. Depending on the low-level task availability, the CADDY compliant mission control system will properly select which high-level functionalities can be activated allowing, in turn, the enabling of the required primitives to fulfil the mission operations.

For the automatic selection, activation and inter-task conflict management, a Petri net based execution control system has been developed. The system is configured by means of a set of configuration files that specify, on one side, the capabilities of the robot in terms of autonomous tasks and, on the other side, the set of high level functionalities that the CADDY system has to provide for the diver support. A real-time Petri net engine models the logical interconnections among the tasks and primitives and, depending on the specific actions commanded by the diver, automatically handle the activation/deactivation of the proper task sets.

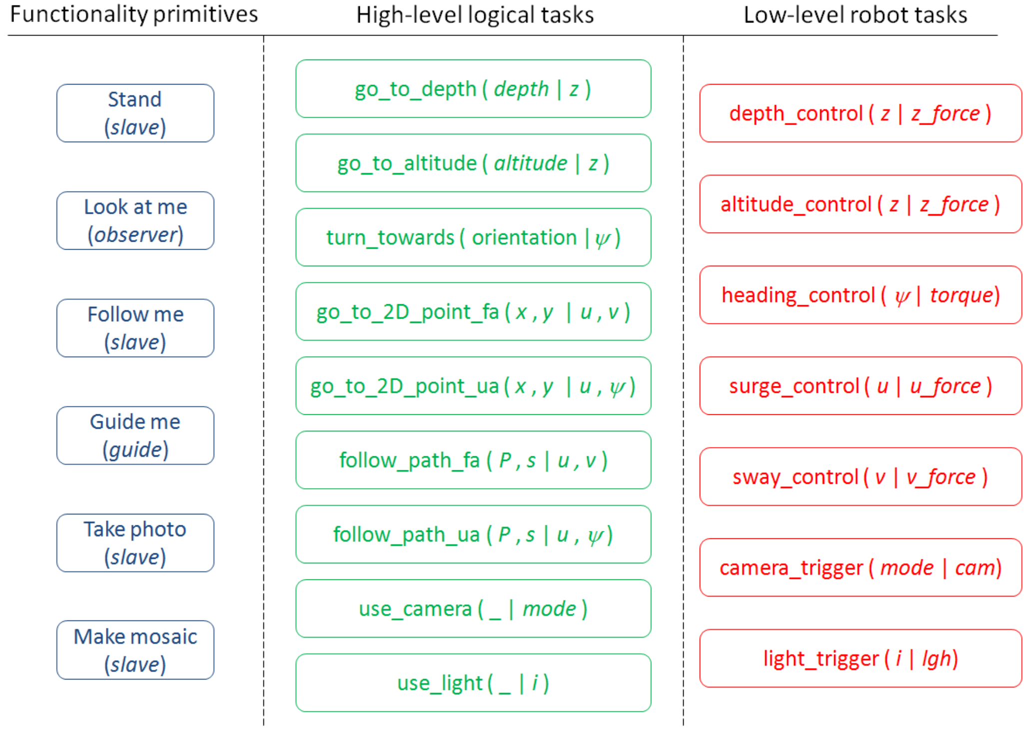


Figure 7. Primitives and tasks definition

6 Validation trials

6.1 Experiment description

Two trials were performed in order to validate the performance of the overall symbolic language interpreter pipeline, one in Biograd na Moru, Croatia in October 2015 and other in Genoa, Italy one month later in November 2015.

During the Biograd na Moru trials, the first approaches towards the gesture classification task were tested, and the underwater videos collected were used for further processing and fine-tuning of the classification algorithms. Mainly, during this period, the slang version of the CADDIAN language was tested.

For this reason, further trials in Genoa, Italy in November 2015 were organized. In this way, not only the slang but the complex gestures were tested. Besides, this also offers the opportunity to test the system in different underwater environments (type of water and weather conditions) and gather feedback from different divers that learn the CADDIAN language. Figure 8 shows different pictures during the data collections and system validation, which includes an introduction to the CADDIAN language for the divers.

All data collection was done without problems using the equipment described above; just for every location, a calibration map to rectify the stereo images was tuned. It is important to mention that in these data collection events only static gestures were analyzed; a list of these is presented next.

- Slang: Go up, go down, go backwards, take a photo, carry equipment.
- Single static gestures: Numbers 1-5, number delimiter, start communication, end communication, do a mosaic, go to the boat, here.
- Complex gestures:
  - Do a mosaic of NxM meters.
  - Go up/down/backwards N meters and take a photo.
  - Go to the boat and carry the equipment here.

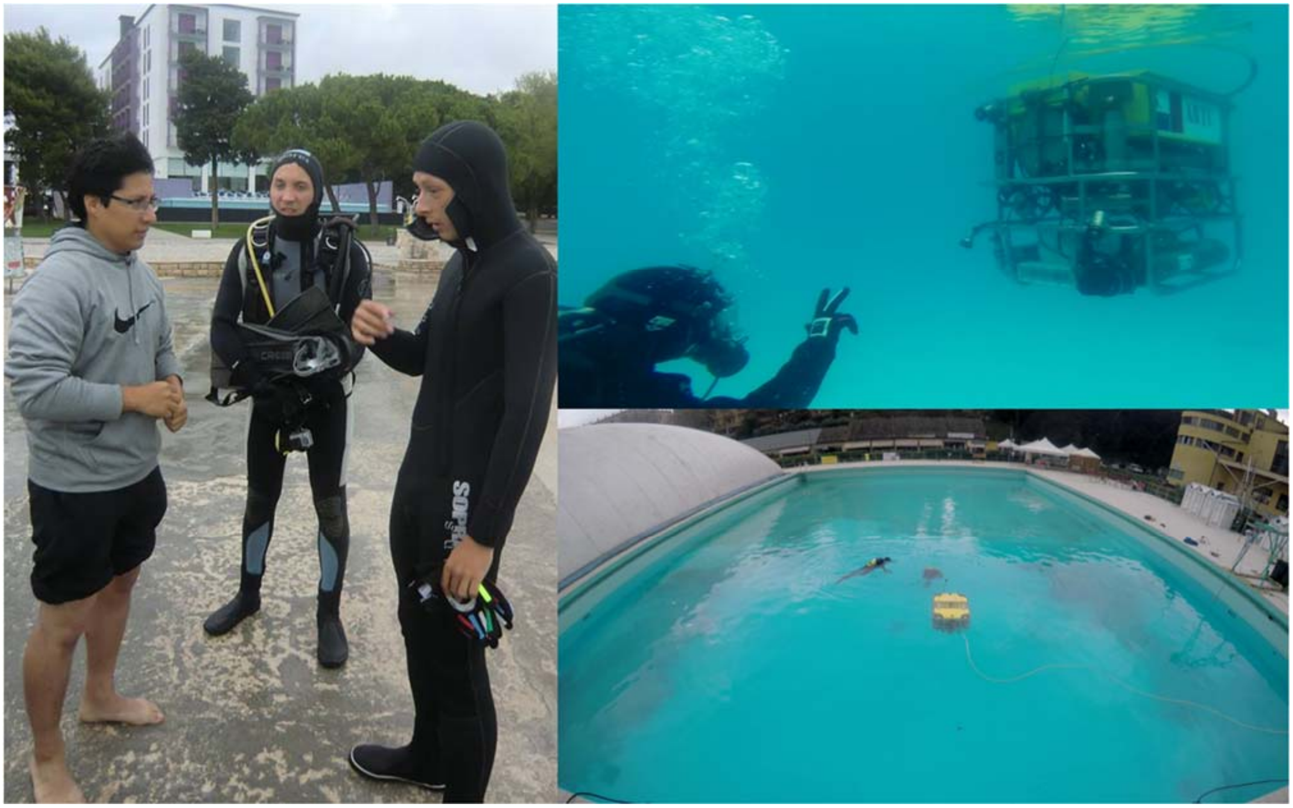


Figure 8. Data collection and testing in Biograd na Moru, Croatia (Left) and Genoa, Italy. This includes the teaching process of the CADDIAN language to the divers.

The next images show the divers performing the gestures in different type of environments to emphasize the different challenges which need to be solved to output a correct classification. For example, the first image (Figure 9) shows a diver facing the light source and a diver giving his back to it. This causes the colors to have less contrast and more attenuation, it can be seen that the red color of the gloves in the left image is close to be black.

The second image (Figure 10), shows how the diver can vary greatly his relative pose with respect to the camera; which in consequence causes the hand's pose and size to change. This happened during the Biograd na Moru trials when the divers could not keep a static position due to the weather conditions that created strong waves in the sea.

The third image (Figure 11) shows also a difference of the diver's position regardless of the fact that the same gesture is performed. Furthermore, this also proves that the localization of the hands cannot depend on the localization of other parts of the body e.g. the head, which is commonly done in other standard RGB-D gesture recognition approaches. In this image, we can also see the difference in color with respect to the previous ones: it seems to have more blue component. This last image is from the Genoa trials, and the previous ones from Biograd na Moru; the different types of water (salt/sweet), weather conditions and background imagery cause these changes in color.

All of the previous proves that the classifier has to be robust against rotations, changes in illumination and different scales. It is difficult for one type of descriptor to be invariant to all these image changes: for this reason a multi-descriptor classifier was adopted as mentioned.

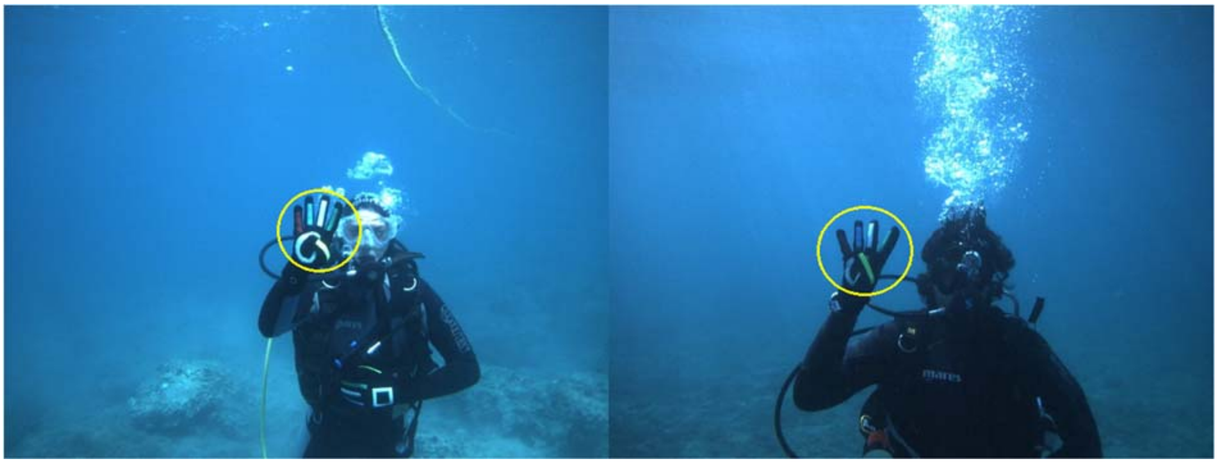


Figure 9. An example from field experiment: depending on the light conditions, colour markers on the gloves are differently interpreted.

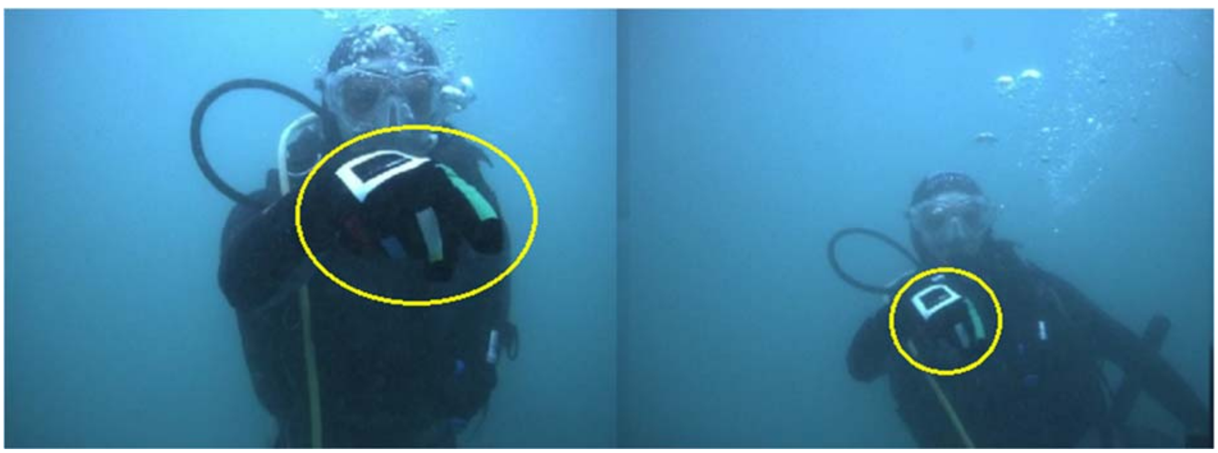


Figure 10. An example from a field experiment where the diver is changing pose while showing the same gesture.

Figure 11. An example from field experiment: divers find it difficult to keep the same pose while issuing a command.

6.2 Result statistics

In order to quantify the results of the classification algorithm, Table 1 shows the classification rate for the different type of individual gestures; it is the accuracy of the hierarchical approach described (Haar+Md-NCMF) using the classification of every frame. Each gesture was tested 10 times using 5 different divers; for a gesture to be considered stable, it has to have the same classification for 48 consecutive frames, which is approximately 3 seconds according to the stereo camera Bumblebee XB3. For these 10 attempts we registered the number of times the algorithm misclassified the gesture when the taking into account the “stability” factor.

Gesture	Classif. Rate per frame	No. misses
Number 1	0.960417	0
Number 2	0.92083	1
Number 3	0.839583	2
Number 4	0.8583	2
Number 5	0.8083	3
Start comm.	0.889583	2
End comm.	0.86875	2
Do mosaic	0.797917	3
Number delimiter	0.90625	1
Go up*	0.98125	0

Go down*	0.9583	1
Go backwards*	0.864583	2
Take a photo*	0.9125	2
Here	0.88125	1
Carry equipment*	0.96667	0
Go to the boat	0.952083	0

*Table 1. Performance parameters for the hand gesture classifier. Gestures marked with * referred to the Slang CADDIAN gestures*

As it can be seen in the previous table, all of the gestures have a classification rate from equal or above 80% approximately; which for number the number of classes and type of imagery is a more than acceptable result according to state of the art in hand recognition. The major problem is between the gesture “Number 5” and “Do mosaic”, which are similar gestures if we consider that the full palm is shown to the camera and all the fingers are extended. The difference is that the first one shows the front of the hand and the second one the back; also the “Do mosaic” command uses both hands. Thus, there is room for improvement if more information is considered like the number of visible hands per frame and pose.

Also, all of the complex gestures were detected and correctly evaluated by the syntax checker; if the diver performed a mistake when giving a gesture or if the classifier misclassified a single gesture from the chain of commands the syntax checker was able to detect the logical error in the sequence and reject it. This prevents the underwater vehicle to perform unnecessary or unfeasible tasks. The next table shows the number of times a complex gesture was rejected by the syntax checker due to errors in the classifier (diver errors were not considered); every complex sequence was performed 15 times.

Complex Gesture	Number of logical rejections and %	Inaccurate classification of parameters	Overall number of errors and error %
Do a mosaic of NxM meters	6 - 40%	1	6 - 40%
Go up N meters and take a photo	0 - 0%	1	1 - 6.7%
Go down N meters and take a photo	0 - 0%	2	2 - 13.3%
Go backwards N meters and take a photo	2 - 13.3%	2	3 - 20%
Go to the boat and carry equipment here	2 - 13.3%	0	2 - 13.3%

Table 2. Performance of the classification of complex gestures out of 15 attempts

Table 2 shows the number of logical rejections output by the syntax checker, also the number of times the parameters of the complex command were misclassified i.e. the numerical parameters. Finally, it shows the overall number of errors made, either logical or parametric. It can be appreciated that the number of logical errors is low overall, except for the "Do mosaic of NxM meters", which means that the command's final objective is almost always interpreted correctly by the system. In case of parametric error, the mission will be carried out because the numerical parameters are always within the scope of the vehicle's capabilities. If we consider the errors made while interpreting the parameters, the overall error is still below 20% (3 out of 15). Some of the parametric errors and the "Do mosaic of NxM meters" can be explained with the Table 1; this shows that the number 5 is commonly mislabeled as the "Do mosaic" command, which will then output a logical error as indicated in Table 2. However, these logical errors will not be executed by the vehicle. The highest priority is the integrity of the overall system i.e. safety of the diver and the AUV, for this reason commands are rejected unless there is great confidence that the mission is feasible.

7 Conclusions

Overall, we can establish that the validations trials were successful because communication between the diver and the AUV was established in a robust manner; which can cope with different environmental conditions (light, weather) and specific personal traits of the divers (preference to perform the gestures vertically or horizontally). But this further points are important:

- The classification method is robust against the different image distortions present in underwater imagery which is a major concern in UAV applications that use cameras.
- The classification of individual gestures presents good accuracy rates, average of 89.8%. The average accuracy rate for the slang gestures (one single gesture commands) is 92.5%. However, the correct interpretation of sequence of commands is of more importance to be able to communicate missions with several number of parameters and subtasks. The average accuracy of these complex commands is 81.3%, and we expect more challenges as the number of gestures increases.
- Based on the previous point, a new approach to make the overall classification of a series of gestures needs to be proposed; the integrity of the mission cannot rely in the misclassification of one single gesture. For example, a Markov process that suggests the probability of a gesture based on the previous ones could help the system be more accurate or have the capability of correcting the misclassification of single gestures in the sequence; similar to how speech-recognition programs work now.
- The mission controller for the AUV controls the navigation system correctly and under bad weather conditions given that GPS readings are available.

8 Annexes

8.1 Annex A - Code gestures table

GESTURE	CODE
#NUMBERS	
GESTURE_NUMBER_ONE	1
GESTURE_NUMBER_TWO	2
GESTURE_NUMBER_THREE	3
GESTURE_NUMBER_FOUR	4
GESTURE_NUMBER_FIVE	5
GESTURE_NUMBER_SIX	6
GESTURE_NUMBER_SEVEN	7
GESTURE_NUMBER_EIGHT	8
GESTURE_NUMBER_NINE	9
GESTURE_NUMBER_ZERO	0 # same as GESTURE_WORK_WAIT
#CADDIAN DELIMITERS	
GESTURE_CADDIAN_START_COMM	10
GESTURE_CADDIAN_END_COMM	11
GESTURE_CADDIAN_NUMBER_DELIMITER	12
#CADDIAN AGENTS	
GESTURE_AGENT_YOU	24 # same as GESTURE_MOV_GO_BACKWARD
GESTURE_AGENT_ME	13
GESTURE_AGENT_WE	14
#QUANTITY	
GESTURE_QUANTITY_PLUS	21 # same as GESTURE_MOV_GO_UP
GESTURE_QUANTITY_MINUS	22 # same as GESTURE_MOV_GO_DOWN
#ACTIONS	
GESTURE_MOV_GO_UP	21 # same as GESTURE_QUANTITY_PLUS
GESTURE_MOV_GO_DOWN	22 # same as GESTURE_QUANTITY_MINUS
GESTURE_MOV_GO_FORWARD	23
GESTURE_MOV_GO_BACKWARD	24 # same as GESTURE_AGENT_YOU
#PLACES	

GESTURE_PLACE_HERE	27
GESTURE_PLACE_BOAT	28
GESTURE_PLACE_POINT_OF_INTEREST	29
#SET VARIABLES	
GESTURE_SET_VAR_AIR	31
GESTURE_SET_VAR_LIGHT	32
GESTURE_SET_VAR_LEVEL	33
GESTURE_SET_VAR_SPEED	34
#WORKS	
GESTURE_WORK_WAIT	0
GESTURE_WORK_MOSAIC	20
GESTURE_WORK_TAKE_PHOTO	25
GESTURE_WORK_CARRY	26
GESTURE_WORK_CHECK	35
GESTURE_WORK_TURN	36
GESTURE_WORK_FOR	37
GESTURE_WORK_END_FOR	38
GESTURE_WORK_DO	39
GESTURE_WORK_FOLLOW	50
GESTURE_WORK_TAKE	51
GESTURE_WORK_COME	52
#LEVEL	
GESTURE_LEVEL_FREE	71
GESTURE_LEVEL_CONST	72
GESTURE_LEVEL_LIMIT	73
#FEEDBACK	
GESTURE_FEED_OK	81
GESTURE_FEED_NO	82
GESTURE_FEED_QUESTION	83
# ONLY SLANG	
GESTURE_SLANG_LOW	84
GESTURE_SLANG_RESERVE	85
#EMERGENCY	
GESTURE_EMERGENCY_DANGER	40
GESTURE_EMERGENCY_OUT_OF_AIR	41
GESTURE_EMERGENCY_OUT_OF_BREATH	42
GESTURE_EMERGENCY_GENERAL_EVACUATION	43
GESTURE_EMERGENCY_PROBLEM	44
GESTURE_EMERGENCY_COLD	45
GESTURE_EMERGENCY_CRAMP	46

GESTURE_EMERGENCY_EAR	47
GESTURE_EMERGENCY_VERTIGO	48
GESTURE_EMERGENCY_ABORT_MISSION	49

8.2 Annex B – Commands, Sequence, semantics table

COMMAND	SEQUENCE	SEMANTICS for Mission Controller
# ear problem		
	[10,47,10]	EM_EAR
	[10,47,11]	EM_EAR
# out of air		
	[10,41,10]	EM_OUT_OF_AIR
	[10,41,11]	EM_OUT_OF_AIR
# I'm cold		
	[10,45,10]	EM_COLD
	[10,45,11]	EM_COLD
# out of breath		
	[10,42,10]	EM_OUT_OF_BREATH
	[10,42,11]	EM_OUT_OF_BREATH
# not ok		
	[10,44,10]	EM_NOT_OK
	[10,44,11]	EM_NOT_OK
# danger		
	[10,40,10]	EM_DANGER
	[10,40,11]	EM_DANGER
# cramp		
	[10,46,10]	EM_CRAMP
	[10,46,11]	EM_CRAMP
# vertigo		
	[10,48,10]	EM_VERTIGO
	[10,48,11]	EM_VERTIGO
# take me to the boat		
	[10,28,10]	GO_TO_BOAT
	[10,28,11]	GO_TO_BOAT
	[10,24,51,13,28,10]	ROBOT TAKE DIVER BOAT
	[10,24,51,13,28,11]	ROBOT TAKE DIVER BOAT
# take me to point of interest		
	[10,24,51,13,29,10]	ROBOT TAKE DIVER POINT_OF_INTEREST

	[10,24,51,13,29,11]	ROBOT TAKE DIVER POINT_OF_INTEREST
# come/return here		
	[10,27,10]	COME_HERE
	[10,27,11]	COME_HERE
	[10,24,52,27,10]	ROBOT COME HERE
	[10,24,52,27,11]	ROBOT COME HERE
# come/return boat		
	[10,28,10]	GO_TO_BOAT
	[10,28,11]	GO_TO_BOAT
	[10,24,52,28,10]	ROBOT COME BOAT
	[10,24,52,28,11]	ROBOT COME BOAT
# come/return point of interest		
	[10,24,52,29,10]	ROBOT COME POINT_OF_INTEREST
	[10,24,52,29,11]	ROBOT COME POINT_OF_INTEREST
# go up [default] or go up 12m		
	[10,21,10]	GO_UP_SLANG
	[10,21,11]	GO_UP_SLANG
	[10,21,1,2,12,10]	GO_UP 12
	[10,21,1,2,12,11]	GO_UP 12
# go down [default] or go down 12m		
	[10,22,10]	GO_DOWN_SLANG
	[10,22,11]	GO_DOWN_SLANG
	[10,22,1,2,12,10]	GO_DOWN 12
	[10,22,1,2,12,11]	GO_DOWN 12
# go forward 565m		
	[10,23,5,6,5,12,10]	GO_FORWARD 565
	[10,23,5,6,5,12,11]	GO_FORWARD 565
# go backward 565m		
	[10,24,5,6,5,12,10]	GO_BACKWARD 565
	[10,24,5,6,5,12,11]	GO_BACKWARD 565
# I follow you		
	[10,13,50,24,10]	DIVER FOLLOW ROBOT
	[10,13,50,24,11	DIVER FOLLOW ROBOT
# You follow me		
	[10,24,50,13,10]	ROBOT FOLLOW DIVER
	[10,24,50,13,11]	ROBOT FOLLOW DIVER

# Stop []interruption of action]		
	[10,82,10]	NO
	[10,82,11]	NO
	[10,24,82,39,10]	ROBOT NO DO
	[10,24,82,39,11]	ROBOT NO DO
# Continue [continue previous action]		
	[10,24,81,39,10]	ROBOT OK DO
	[10,24,81,39,11]	ROBOT OK DO
	[10,24,39,10]	ROBOT DO
	[10,24,39,11]	ROBOT DO
	[10,81,10]	OK
	[10,81,11]	OK
# general evacuation		
	[10,43,11]	EM_GEN_EVACUATION
	[10,43,11]	EM_GEN_EVACUATION
#abort mission		
	[10,49,11]	EM_ABORT_MISSION
	[10,49,11]	EM_ABORT_MISSION
#slow down		
	[10,34,22,10]	SPEED MINUS
	[10,34,22,11]	SPEED MINUS
#accelerate		
	[10,34,21,10]	SPEED PLUS
	[10,34,21,11]	SPEED PLUS
#keep this level		
	[10,33,72,10]	LEVEL LEVEL_CONST
	[10,33,72,11]	LEVEL LEVEL_CONST
	[10,72,10]	LEVEL_CONST
	[10,72,11]	LEVEL_CONST
#free level		
	[10,33,71,10]	LEVEL LEVEL_FREE
	[10,33,71,11]	LEVEL LEVEL_FREE
#level off		
	[10,33,73,10]	LEVEL LEVEL_LIMIT
	[10,33,73,11]	LEVEL LEVEL_LIMIT

# set point of interest		
	[10,29,10]	SET_POINT_OF_INTEREST
	[10,29,11]	SET_POINT_OF_INTEREST
# give me light		
	[10,32,21,10]	LIGHT PLUS
	[10,32,21,11]	LIGHT PLUS
# no more light		
	[10,32,22,10]	LIGHT MINUS
	[10,32,22,11]	LIGHT MINUS
# give me air		
	[10,31,21,10]	AIR PLUS
	[10,31,21,11]	AIR PLUS
# no more air		
	[10,31,22,10]	AIR MINUS
	[10,31,22,11]	AIR MINUS
# no		
	[10,82,10]	NO
	[10,82,11]	NO
# ok		
	[10,81,10]	OK
	[10,81,11]	OK
# I do not understand		
	[10,83,10]	DONT_KNOW
	[10,83,11]	DONT_KNOW
# Mosaic of 25 * 11 area		
	[10,20,2,5,12,1,1,12,10]	MOSAIC 25 11
	[10,20,2,5,12,1,1,12,11]	MOSAIC 25 11
# Mosaic of 33 * 33 area		
	[10,20,3,3,12,10]	MOSAIC 33
	[10,20,3,3,12,11]	MOSAIC 33
# Take photo of 25 * 11 area		
	[10,25,2,5,12,1,1,12,10]	TAKE_A_PHOTO 25 11
	[10,20,2,5,12,1,1,12,11]	TAKE_A_PHOTO 25 11
# Take photo of 33		
	[10,20,3,3,12,10]	TAKE_A_PHOTO 33
	[10,20,3,3,12,11]	TAKE_A_PHOTO 33

# Wait 5 seconds/minutes		
	[10,0,5,12,10]	WAIT 5
	[10,0,5,12,11]	WAIT 5
# Check,tell me what you are doing		
	[10,35,10]	CHECK
	[10,35,11]	CHECK
# carry a tool for me		
	[10,26,10]	CARRY
	[10,26,11]	CARRY
# stop carrying a tool for me		
	[10,82,26,10]	NO CARRY
	[10,82,26,11]	NO CARRY
# turn 180°		
	[10,36,10]	TURN
	[10,36,11]	TURN
# for cycle [example]		
	['10', '37', '1', '2', '12', '26', '20', '3', '3', '12', '25', '29', '82', '26', '0', '5', '12', '38', '10']	FOR 12 TIMES DO CARRY MOSAIC 33 TAKE_A_PHOTO POINT_OF_INTEREST NO CARRY WAIT 5 END_FOR
	['10', '37', '1', '2', '12', '26', '20', '3', '3', '12', '25', '29', '82', '26', '0', '5', '12', '38', '11']	FOR 12 TIMES DO CARRY MOSAIC 33 TAKE_A_PHOTO POINT_OF_INTEREST NO CARRY WAIT 5 END_FOR
# where is the boat		
	[10,83,28,10]	WHERE_IS_THE_BOAT?
	[10,83,28,11]	WHERE_IS_THE_BOAT?
# are you ok?		
	[10,83,44,10]	ARE_YOU_OK?
	[10,83,44,11]	ARE_YOU_OK?
# how much air do you have letf?		
	[10,83,31,10]	HOW_MUCH_AIR_DO_YOU_HAVE_LEFT?

	[10,83,31,11]	HOW_MUCH_AIR_DO_YOU_HAVE_LEFT?
# low on air		
	[10,84,10]	LOW_ON_AIR
	[10,84,11]	LOW_ON_AIR
#on reserve		
	[10,85,10]	ON_RESERVE
	[10,85,11]	ON_RESERVE