Robust Multimodal Command Interpretation for Human-Multirobot Interaction

Jonathan Cacace, Alberto Finzi, and Vincenzo Lippiello

Università degli Studi di Napoli Federico II
{jonathan.cacace,alberto.finzi,lippiello}@unina.it

Abstract. In this work, we propose a multimodal interaction framework for robust human-multirobot communication in outdoor environments. In these scenarios, several human or environmental factors can cause errors, noise and wrong interpretations of the commands. The main goal of this work is to improve the robustness of human-robot interaction systems in similar situations. In particular, we propose a multimodal fusion method based on the following steps: for each communication channel, unimodal classifiers are firstly deployed in order to generate unimodal interpretations of the human inputs; the unimodal outcomes are then grouped into different multimodal recognition lines, each representing a possible interpretation of a sequence of multimodal inputs; these lines are finally assessed to recognize the human commands. We discuss the system at work in a real world case study in the SHERPA domain.

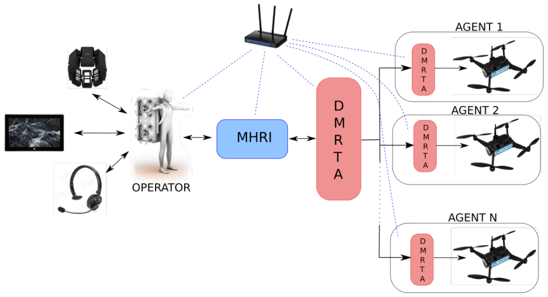
Introduction

In this work, we tackle the problem of robust multimodal communication between a human operator and a team of robots during the execution of a shared task in outdoor environments. In these scenarios, the robots should be able to timely respond to the operator's commands, minimizing chances of misunderstanding due to noise or user errors. This crucial problem is well illustrated by the domain of the SHERPA project [10, 3], whose goal is to develop a mixed ground and aerial robotic platform supporting search and rescue (SAR) activities in an alpine scenario. One of the peculiar aspects of the SHERPA domain is the presence of a special rescue operator, called the *busy genius*, that cooperates with a team of aerial vehicles in order to accomplish search and rescue missions. In this context, the human operator is not fully dedicated to the control of the robots, but also involved in the rescue operations. On the other hand, he/she can exploit light wearable devices to orchestrate the robotic team operations in a multimodal manner, using voice and gestures based commands, in order to enable a fast and natural interaction with the robots. This scenario challenges the command recognition system, since the environment is unstructured and noisy, the human is under pressure, and the commands are issued in a fast and sparse manner. In order to support the operator in similar scenarios, a robust and reliable multimodal recognition system is a crucial component. In multimodal interaction frameworks [8, 2, 4, 7, 12], multimodal fusion is a key issue and different strategies have been proposed [1] to combine the data provided by multiple input channels (gestures, speech, gaze, body postures, etc.). Analogously to [11, 9], in order to make the system interaction robust, extensible, and natural, we adopt a late fusion approach where the multimodal inputs provided by the human are first processed by dedicated unimodal classifiers (gestures recognition, speech recognition, etc.) and then recognized by combining these outcomes. In this setting, multimodal data are usually first synchronized and then interpreted according to rules or other classification methods. In contrast with these solutions, in this work we propose a novel multimodal fusion approach that avoids explicit synchronization among incoming multimodal data and it is robust with respect to several sources of errors, from human mistakes (e.g. *delays in utterances or gestures*, *wrong and incomplete sequencing*, *etc.*) and environmental disturbances (e.g. *wind, external noises*), to unimodal classification failures. The main idea behind the approach is to continuously assess multiple ways to combine together the incoming multimodal inputs in order to obtain a subset of events that better represent a human multimodal command. In particular, command recognition is performed in two decision steps. In the first one, we generate multiple hypothesis on multimodal data association given a Bayesian model of the user way of invoking commands. For this purpose, we estimate the probability that new samples are related to others already received. Then, in a second step, a *Naive Bayes* classifier is deployed to select the most plausible command given the possible data associations provided by the previous step.

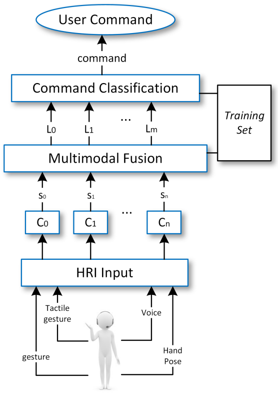
Multimodal Human-Robot Interaction Architecture

In Figure 1(a) we illustrate the human-multirobot architecture. The human operator interacts with the robotic platform using different communication channels (*i.e. Voice*, *Arm Gestures*, *Touch Gestures* and *Hand Poses*) by means of his/her wearable devices. In particular, the operator exploits a headset to issue vocal commands, a motion and gesture control brand (*Myo Thalmic Armband*[1]) and a mobile device (*tablet*) with a touch based user interface. The multimodal interaction system (*MHRI*) should then interpret these commands passing them to the Distributed Multi-Robot Task Allocation (*DMRTA*) (see [5] for details). In this work, we focus on the *MHRI* describing the multimodal command recognition system illustrated in Figure 1(b). Raw device data are directly sent and simultaneously elaborated by the unimodal classifiers $C_0, ..., C_n$ in order to generate the unimodal samples s_i. These samples are then received by the *Multimodal Fusion* module to generate different recognition lines $\{L_0, ..., L_m\}$ exploiting the Bayesian Network and the *Training Set*. Each multimodal command is successively interpreted as a user command by the *Command Classification* module.

[1] https://www.myo.com/



(a) Human-Robot Interaction architecture.



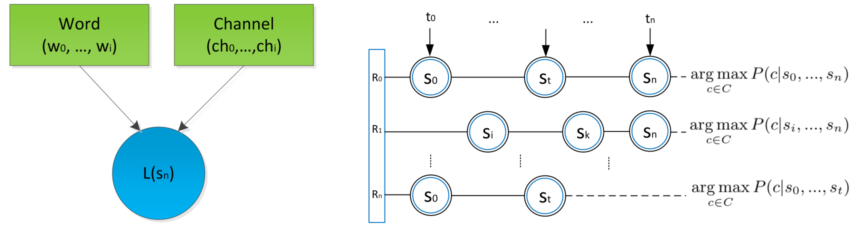
(b) Multimodal Recognition System.

Command Recognition

Multimodal command recognition relies on a late fusion approach in which heterogeneous inputs provided by the user through different channels are first classified by unimodal recognizers and then fused together in order to be interpreted as human commands. More specifically, given a sequence of inputs S generated by the unimodal classifiers, the command recognition problem consists in finding the command c that maximizes the probability value $P(c|S)$. This problem is here formulated as follow. We assume a set $C = \{c_0, c_1, ..., c_k\}$ of possible commands invokable by the operator. Each command is issued in a multimodal manner, hence it is associated with a sequence of unimodal inputs $S = \{s_0, ..., s_n\}$, each represented by the triple $s_i = (w_i, ch_i, t_i)$, where $w_i \in W$ is the label provided by the unimodal classifier associated with the $ch_i \in I$ channel and $t_i \in \mathbb{R}^+$ is its time of arrival. In our approach, the user commands are interpreted in two decision steps: firstly, the outputs of unimodal classifiers are fused together (*Multimodal Fusion*) in order to be assessed an recognized as user commands in the second step (*Command Recognition*).

Multimodal Fusion. The multimodal fusion step allows the system to select and group together unsynchronized inputs provided by the unimodal classifiers associated with the same current command. For this purpose, in correspondence to the input sequence S of unimodal classified data, we generate different possible subsets of elements, called *Recognition Lines*, each representing a possible way to associate these inputs to the invoked command. Therefore, during the command interpretation process, different *Recognition Lines* are generated and collected into a *Recognition Set* in order to be interpreted in the second step. These multiple ways of grouping the inputs, allow the proposed framework to fuse unsynchronized unimodal inputs in a robust fashion, coping with disturbances like environmental noise, command invocation errors, or failures of the single unimodal recognition subsystems. The *Recognition Line* generation process works as follows. First of all, for each new input, a new *Recognition Line* containing only this data is generated; then the incoming data is also assessed in order to be included into other *Recognition Lines* already available in the *Recognition Set*. In order to assign an input sample to a *Recognition Line* we rely on a Bayesian Network (BN) approach suitably trained in order to infer the probability that a new incoming unimodal sample s_n belongs to a *Recognition Line* given the others $s_0, ..., s_i$ already associated to the same line. Specifically, the BN proposed in this work consists of three different nodes (see Figure 1(c)). The *Word* node, that contains the list of input data in the recognition line, the *Channel* node that is for the input channels and the *Line* node that represents the probability of the new incoming samples to belong to the considered line. In this setting, a recieved input data is associated with a recognition line the probability to belong to that line is greater than a suitable threshold τ_1 and the temporal distance of the received sample (s_r) with respect to the previous one (s_p) on the same line is within a specific interval ($|t_{s_r} - t_{s_p}| < \gamma$).

Command Recognition. In the command recognition phase, the previously generated *Recognition Lines* are to be interpreted as user commands. Our approach exploits a *Naive Bayes* classifier to associate each element of the *Recognition Set* with a label and a score representing, respectively, the recognized command class and its classification probability. More specifically, given a sequence of samples $S = \{s_0, ..., s_n\}$, the list of *semantic* labels $S_w = \{w_0, ..., w_n\}$ is extracted. Given the list of possible commands $c_0, ..., c_k$, the class $\hat{c}$ and its score is assessed by through the formula: $\hat{c} = \arg\max_{c \in C} P(c) \prod_{i=1}^{|S_w|} p(c|w_i)$. Once all the *Recognition Lines* have been classified, the line with maximum score is selected as the recognized user command (see Figure 1(d)). Also in this case, a command is properly recognized only if the probability returned by the *Naive Bayes* classifier is higher than a trained threshold τ_2 .


Word
(w₀, ..., wᵢ)
Channel
(ch₀,...,chᵢ)
L(sₙ)
t₀
...
...
tₙ
R₀
R₁
Rₙ
S₀
Sₜ
Sₙ
Sᵢ
Sₖ
$\arg\max_{c \in C} P(c|s_0, ..., s_n)$
$\arg\max_{c \in C} P(c|s_i, ..., s_n)$
$\arg\max_{c \in C} P(c|s_0, ..., s_t)$

(c) *Bayesian Network* for multimodal command fusion.

(d) Recognition lines and scores.

System Training. The multimodal system is trained exploiting a *Training Set* that collects, for each sample: the requested command coupled with the generated samples, the associated channel, and the elapsed time between the samples.

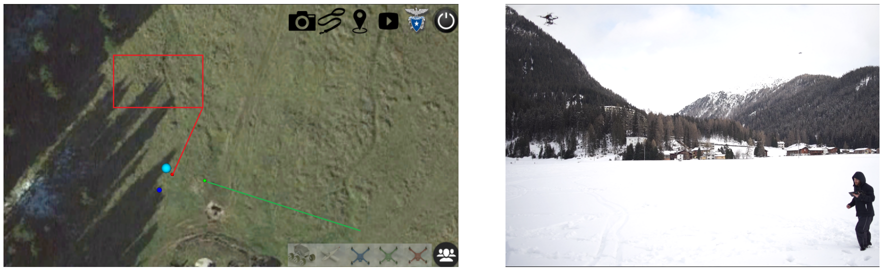
This way, the *Bayesian Network* for *multimodal fusion* is trained by with list of pairs (w_i, ch_i) for each command invocation in the dataset. The *command recognition system* is trained with the list of (w_i) of the samples used to interpret the user commands. Moreover, once the *multimodal fusion* system has been trained, a final training session is needed to adapt the thresholds (τ_1, τ_2, γ). This is obtained by asking the users to validate both the generated *Recognition Lines* and the associated classification result.

SHERPA Case Study

The proposed system has been demonstrated and tested in a real Alpine environment. In order to communicate with the robotic platforms, the operator is equipped with wearable devices: a standard *headset*, a mobile device (*tablet*) along with a gesture/motion control bracelet. Speech recognition is based on the *PocketSphinx*[2] software adopting a *bag-of-words* model instead of the most commonly used context-free grammars. Grammar based models exploit the word ordering in the sentence, which is not reliable in our setting since the user can accidentally skip words because the interaction is sparse and incomplete or the recognizer fails to catch words, because the environment is noisy. In contrast, we adopt a less restrictive model where the recognized sentences are represented as bags of words, which are then further processed in the late fusion step of the multimodal recognition system described above. Gesture based commands are used to control the robotic team via complete or complementary information (*i.e. pointing or mimic gestures*). We designed and implemented a continuous gesture recognition module based on the approach by [13]. Gesture classification is here based on the acceleration of the operator's arm, which is detected by a lightweight *IMU*-based bracket. We defined 14 different types of gestures used to invoke high level actions (i.e. *direction movements, circles, approaching, etc*). These gestures have been trained using a data set that collects gestures from 30 users, each providing 10 trials of each gesture class. The operator is also able to issue commands by drawing *2D* gestures on a touch *user interface* (see Figure 1(e)). In this case, areas to explore, trajectories or target points can be specified using geometrical shapes like *Circles*, *Squares* or *Lines* eventually paired with voice information. The operator can also specify commands or part of them using hand poses. The hand pose recognition system is implemented exploiting the built-in *Myo Armband* classifier able to discriminate five different hand poses from EMG sensors, namely *double-tap*, *spread*, *wave left*, *Wave Right* and *Make Fist*. As for the user dataset, we mainly focus on commands suitable for interacting with a set of co-located drones during navigation and search tasks. Namely, *selection* commands enable the operator to select single or groups of robots; for this purpose the operator can issue speech (e.g. *all drones take off*, *red drone land*), speech and gestures in combination (e.g. *you go down*), including touch gestures for the *user interface*. Similar combination of modalities

[2] http://wiki.ros.org/pocketsphinx

can be exploited to invoke *motion* and *search* during navigation and exploration tasks.



(e) Touch Screen User Interface. In *red* an area to explore, in *green* a path to navigate.

(f) Human operator interacting with multiple drones in a snow-clad field.

System Training. The overall system requires three training sessions. The first one is related to the unimodal classifiers set up. The second training phase concerns the multimodal fusion engine. It requires the *Training Set* introduced above, exploited by the system to learn how the operator generates commands, that is, how he/she composes the unimodal samples to invoke commands. Notice that in our scenario the operator is an expert rescuer already aware about the system and the operative domain, therefore we trained the system with 4 trained users (involved in the research project), asking them to repeat 45 commands 10 times each. The collected data are then used to train both the *multimodal fusion* and the *command recognition* system. A final training phase is needed to tune the τ_1 and τ_2 thresholds.

System Testing. The robotic platform set up and the scenario is analogous to the one described in [5]. The testing site is the one depicted in Figure 1(f). In this context, we collected data from *14* different missions lasting about *15* minutes each and performed in two different days. A more extended description and discussion of these tests can be found in [6], here we only summarize the main results about the system robustness with noisy communication. Specifically, we collected data about 107 commands (and 708 samples) achieving a success rate of 96.8%, even though more than half of the samples generated by the user have been marked as mistakes and rejected by the multimodal fusion algorithm (66.9% rejected samples), among these, 74.3% have been correctly rejected in the recognition line exploited for multimodal classification.

Acknowledgement

The research leading to these results has been supported by the FP7-ICT-600958 SHERPA, ERC AdG-320992 RoDyMan and H2020-ICT-731590 REFILLs projects respectively. The authors are solely responsible for its content. It

does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

References

1. Atrey, P.K., Hossain, M.A., El Saddik, A., Kankanhalli, M.S.: Multimodal fusion for multimedia analysis: a survey. Multimedia Systems 16(6), 345–379 (2010)
2. Bannat, A., Gast, J., Rehrl, T., Rösel, W., Rigoll, G., Wallhoff, F.: A multimodal human-robot-interaction scenario: Working together with an industrial robot. In: Human-Computer Interaction. Novel Interaction Methods and Techniques, 13th International Conference, HCI International 2009, San Diego, CA, USA, July 19-24, 2009, Proceedings, Part II. pp. 303–311 (2009)
3. Bevacqua, G., Cacace, J., Finzi, A., Lippiello, V.: Mixed-initiative planning and execution for multiple drones in search and rescue missions. In: Proceedings of the Twenty-Fifth International Conference on International Conference on Automated Planning and Scheduling. pp. 315–323. ICAPS'15, AAAI Press (2015)
4. Burger, B., Ferrané, I., Lerasle, F., Infantes, G.: Two-handed gesture recognition and fusion with speech to command a robot. Autonomous Robots 32(2), 129–147 (2012)
5. Cacace, J., Finzi, A., Lippiello, V., Furci, M., Mimmo, N., Marconi, L.: A control architecture for multiple drones operated via multimodal interaction in search rescue mission. In: Proc. of SSRR 2016. pp. 233–239 (Oct 2016)
6. Cacace, J., Finzi, A., Lippiello, V.: A robust multimodal fusion framework for command interpretation in human-robot cooperation. In: 26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2017, Lisbon, Portugal, August 28 - Sept. 1, 2017. pp. 372–377 (2017)
7. Dumas, B., Lalanne, D., Oviatt, S.L.: Multimodal interfaces: A survey of principles, models and frameworks. In: Lalanne, D., Kohlas, J. (eds.) Human Machine Interaction, Lecture Notes in Computer Science, vol. 5440, pp. 3–26. Springer (2009)
8. Holzapfel, H., Nickel, K., Stiefelhagen, R.: Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3d pointing gestures. In: Proc. of ICMI 2004. pp. 175–182. ACM (2004)
9. Lucignano, L., Cutugno, F., Rossi, S., Finzi, A.: A dialogue system for multimodal human-robot interaction. In: Proc. of ICMI 2013. pp. 197–204. ACM (2013)
10. Marconi, L., Melchiorri, C., Beetz, M., Pangercic, D., Siegwart, R., Leutenegger, S., Carloni, R., Stramigioli, S., Bruyninckx, H., Doherty, P., Kleiner, A., Lippiello, V., Finzi, A., Siciliano, B., Sala, A., Tomatis, N.: The sherpa project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments. In: Proc. of SSRR 2012. pp. 1–4 (2012)
11. Rossi, S., Leone, E., Fiore, M., Finzi, A., Cutugno, F.: An extensible architecture for robust multimodal human-robot communication. In: Proc. of IROS 2013. pp. 2208–2213 (Nov 2013)
12. Villani, V., Sabattini, L., Riggio, G., Secchi, C., Minelli, M., Fantuzzi, C.: A natural infrastructure-less human-robot interaction system. IEEE Robotics and Automation Letters 2(3), 1640–1647 (2017)
13. Wobbrock, J.O., Wilson, A.D., Li, Y.: Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In: Proc. of UIST 2007. pp. 159–168. ACM (2007)