2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

An Extensible Architecture for Robust Multimodal Human-Robot Communication

Silvia Rossi, Enrico Leone, Michelangelo Fiore, Alberto Finzi and Francesco Cutugno
Department of Electrical Engineering and Information Technologies
University of Naples “Federico II”, Naples, Italy
email: {silvia.rossi, alberto.finzi, cutugno}@unina.it

Abstract— **Human safety and effective human-robot communication are main concerns in HRI applications. In order to achieve such goals, a system should be very robust, allowing little chance for misunderstanding the user’s commands. Moreover, the system should permit natural interaction reducing the time and the effort needed to achieve tasks. The main purpose of this work is to develop a general framework for flexible and multimodal human-robot communication. The proposed architecture should be easy to modify and expand, adding or modifying input channels and changing the multimodal fusion strategies. In this paper, we introduce our general approach and provide a case study with two modalities (gesture and speech).**

I. Introduction

In order to work with humans, a robotic system should be able to understand the users’ commands and intentions, and to safely interact within a shared workspace. New classes of lightweight robots create future scenarios for automation in industrial settings, where they could share space and activities with their human counterparts. These highlights some difficulties with today’s robots, as they do not adapt well to dynamic environments and do not offer rich human-robot interaction (HRI) possibilities [1]. The design of natural interfaces for the interaction is a crucial issue in effective co-working between humans and robots [2]. This is a key issue also in commercial and home applications. Moreover, a richer way to interact allows the users to be less involved in giving instructions to the robot, instead of focusing on their own activities. In a multimodal interaction context, the users can communicate with the robots using several input channels, called modes, that are analyzed and integrated by the system. Indeed, different modes, more than complementing each other, can offer the same information, introducing redundancy in the system. This redundancy can be useful for different purposes. In noisy environments, the information provided on a channel could be not reliable and redundancy can help reducing errors in the interpretation of the input signals (e.g., in a low light environment speech commands may be more reliable than gestures), hence, the robustness is generally enhanced [3]. Finally, multimodal interaction should be more intuitive. When we talk to each other, we frequently use body gestures to complement speech information, indicating objects or supporting our arguments. This kind of interaction seems more flexible and natural, since users can interact with the system using their favorite modalities.

In this paper, we propose a general architecture supporting multimodal human-robot communication and interaction. The architecture is specifically designed to be easily modified and expanded, adding or modifying input channels, and changing the multimodal fusion strategies, without impacting the rest of the system. Most of the multimodal HRI systems proposed in literature are based on a dominant modality; in contrast, we propose a novel approach based on late fusion classification based which permits flexible combination of modalities. We introduce our methodology and provide a case study considering as input channels gestures and speech. We show that our approach allows to obtain a system which is robust (few misunderstandings of users commands), flexible and intuitive (users can freely combine the modalities).

The rest of this paper is organized as follows: in Section II we present related works on multimodal systems and introduce the main problems and approaches; in Section III we introduce our general architecture, while in Section IV we illustrate a case study and our tests results; finally, in Section V we discuss conclusions and future developments.

II. Background and Related Works

In the field of HCI, multimodal interfaces were introduced for the first time in [4], where objects were created and moved on a screen using voice recognition and finger pointing. In recent years, with the widespread adoption of touch screens and fast speech recognition systems, the possibilities of implementing multimodal information access has received a strong acceleration and interfaces supporting truly multimodal commands are available to everyday users.

In the field of the HRI, many approaches exploit multimodal interaction to improve the human-robot communication. However, many of these systems are characterized by a dominant modality. For example, in [5], Holzapfel et al. developed an architecture for multimodal fusion in a kitchen scenario, where speech is the main input modality and pointing gestures are mainly used for object disambiguation. In [6], Burger et al. propose a system that is able to handle natural artefacts performing 3D gestures using a stereo camera and a bank of collaborative particle filters. Also in this approach the main input modality is speech. When the interpreter needs a complementary gesture for disambiguation, the system performs a fusion with the gesture

Authors are listed in reverse alphabetical order.

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

interpretation results provided on the same time window. Differently from these approaches, we do not assume dominant input modalities and a user should be able to interact by using both multimodal and monomodal commands.

There are several issues to consider for the design of a multimodal system. First of all, a multimodal system receives inputs from various devices that are associated with a single or multiple modalities. From these data, useful features can be extracted. This information should be integrated to produce a single interpretation. Multimodal fusion can be produced at the feature level or at the decision level. In feature level approaches, like in [7] (early fusion), the extracted features are combined into a single vector and then sent to an unit that produces an interpretation. The main advantage of this approach is the possibility to use the correlation of different modalities from the first analysis steps. Moreover, the system needs only a single training phase. Feature level fusion is useful when the modalities are strongly coupled (e.g., voice and lips movements). On the other hand, combining features from less coupled modes can produce a lot of noise, caused by joining uncorrelated information. Finally, since features can be extracted at different moments, it is necessary to synchronize their integration.

In this work we adopt a decision level approach (late fusion) [8] that integrates the information from single modalities after they have been interpreted from a recognizer. This strategy is typical in HCI, because it is easily scalable (modalities can be added or canceled), and more adaptable (each mode can be analyzed in the optimal way, using its own recognition model). However, in this approach training is more complex: the system needs to be trained in different phases (e.g., mode recognizers and the integration unit). Synchronization between different modes is also required because of latencies introduced by the modalities recognizers. There are several ways to obtain the modalities fusion. The approaches can be divided in three broad categories: rule based, estimator based, and classification based. Rule Based Methods use sets of rules to integrate multimodal information, as in [5], [6]. Some of these rules are hand-tuned for the application domain, while others are based on statistical methods (e.g., weighted linear fusion). This method is computationally efficient but has the problem of the optimal weight determination. Usually, rule based systems work fairly well but are very dependent on the application domain. Estimators have been sometimes used in multimodal fusion, as in [9], but they are typically used in feature level fusion systems. Finally, many classifiers have been tested for integrating multimodal information, like Support Vector Machines (SVMs) [10], neural networks [11] and Bayesian models [12]. These methods can obtain excellent results, but need longer training phases than rule based methods. Our approach adopts a late fusion classification-based approach. Specifically, multimodal fusion is obtained by exploiting SVMs in a HRI context. Classification-based late fusion engines are not explored in multimodal HRI system, which are usually designed as rule-based systems. Classification systems, and in particular SVMs, have been proven to be very efficient and to produce better results than rule-based systems or other machine learning algorithms [13]. A similar approach was presented in [14] for emotion recognition in HRI, where all the classifiers for modalities and late fusion are implemented using Bayesian models. [14] provides an homogeneous framework using Bayesian networks, while here, the focus is on the adaptability/ extensibility of the framework and the use of a SVM for fusion.

III. System Architecture

The great interest risen by multimodal interaction, highlighted the need of formalizing the requirements an automated interactive system needs to fulfill to be considered multimodal. This problem was also addressed by the W3C, which established a set of requirements, concerning both interaction design [15] and system architecture [16], formalized as proprieties and theoretical standard multimodal architectures. The system, we propose, is structured in different layers and exploits a multimodal late fusion strategy (see Fig.1). The Modalities Recognizers classify the features extracted from the raw data, provided by the sensors, and create a first list of possible interpretations (see Section III-A). These interpretations are then synchronized and integrated in the Fusion Engine (see Section III-B). The Fusion Engine exploits an integration strategy based on classification.

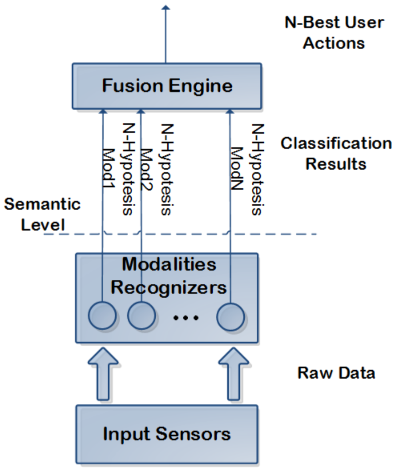

Fig. 1. Architecture for multimodal human-robot communication.

A. Modalities Recognizers

Differently from what happens with mobile devices, in HRI humans can express their intentions using many different channels. Speech and gestures are obviously the most intuitive, however other modalities can semantically contribute to the construction of the intended meaning. For example, robots can use humans' positions, the emotional states (acquired from speech, face expression, body posture, biosensors etc.), further gestures on a touchable device connected to the robot, and so on. According to our architecture, each channel is separately processed to provide elaborated data. The Modalities Recognizers classify the features extracted from the raw data provided by the input sensors and create a first list of possible interpretations (N-Hypothesis) for the fusion engine.

B. Late Fusion Engine

The local decisions are combined in a fused decision vector that is analyzed by another unit to provide a final multimodal interpretation of a command. In the following, we describe the units involved in this process (see Fig.2).

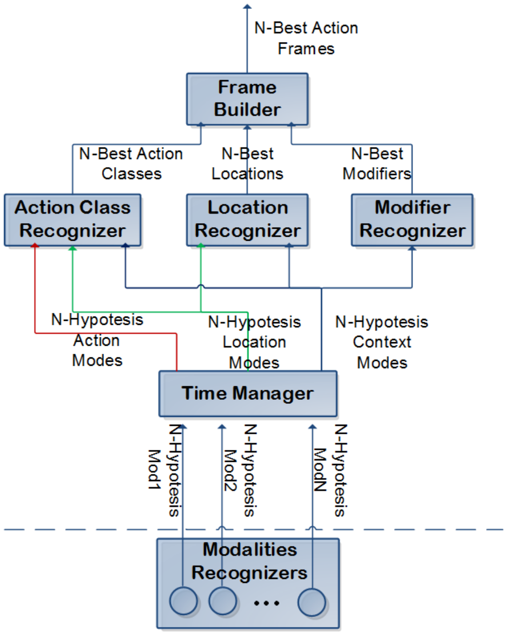

Fig. 2. Fusion Engine.

The *Time Manager* communicates directly with the single-channel recognizers. Its task is to synchronize the inputs of the multiple channels (each message is timestamped, i.e., it contains the starting time and ending time of the data analyzed by the recognizer). This synchronization process is implemented by a set of rules. For example, we consider two signals from different channels to be a part of the same command if they overlap, or one follows another by a fixed delay. The *Action, Object, and Modifier* classifiers receive as input a fused decision vector, composed of the scores of the single modalities recognizers. The *Action classifier* chooses which action the user wants to perform, using additional information about each modality:

- Contextual information, e.g. the amount of noise in the environment, which can help the classifier to differently weight each modality. For example, in a noisy environment, visual information can be considered as more reliable than the audio information.
- Location information, which can help to disambiguate an action, e.g. if the user is pointing to an item while speaking, the classifier can exclude those actions that are not directed toward objects.

The *Object Classifier* chooses where the action should be performed. This can be seen as a regression problem, for modalities whose responses are points with 3D coordinates, like pointing with an arm, or as a classification problem, for modalities whose responses are item labels, like speech commands. The *Modifier classifier* chooses how to perform an action. For example, if the environment is very dark or crowded it could choose to perform the action slowly and carefully. The *Frame Builder* is to build up robot executable actions and to integrate incomplete commands. In our system, commands can be specified sequentially using the same channel. For example, we could ask the robot to take an object by first executing a "take" gesture and then pointing to an object. Notice that the Action Classifier may be not able to integrate these commands, because it takes decisions by combining single interpretations from each modality. After the Frame Builder has received responses from the classifiers, the Frame Builder starts grouping this information. If two commands are incomplete, this module tries to integrate their information thus creating a single complete command, when they are compatible. A command is considered incomplete if it lacks information about one area, like its object or action. Two commands are compatible if each one has information that the other command lacks.

IV. Case Study

In the following we illustrate the previous architecture instantiated in a concrete HRI scenario.

a) Scenario: Many home and industrial settings for HRI requires the possibility for the robot to autonomously navigate (e.g., mobile delivery robots), to be of support for the user during its activities (e.g., tool exchange robots), to physically interact with humans (e.g., cooperative assembly and transportation tasks), otherwise the robots can carry out their tasks (repetitive and requiring an high precision) in the presence of humans. Except for the case of pHRI, some of the common characteristics of these activities can be represented as combinations of pick, place, and carry actions.

As a case study we introduce a scenario where the user and the robot are to cooperate in a lab environment in order to put several colored objects in numbered places. The user can interact with the robot using gestures and speech. Despite the simplicity of this pick-place-carry scenario, we can find enough elements to elaborate several interaction patterns. Indeed, gestures are often ambiguous and may be interpreted in different ways depending on the context and the robot should be able to disambiguate the user commands by integrating the interpretations from different modalities. We designed the interaction introducing a set of primitives, each associated with different combinations of gestures and speech inputs. In particular, we considered the interaction mediated by the following instructions: *Take that object*; *Drop the object*; *Give me the object*; *Stop*; *Re-evaluate the command*; *Go there*; *Search that area*; *Come here*.

b) Input Sensors: The system uses Microsoft Kinect and a microphone as input sensors. Microsoft Kinect is a device features an RGB Camera and a depth sensor. This device allows us to recognize and track different users in a scene representing them with a 20 joints skeleton. These joints do not include the fingers, which are crucial for our gesture recognition problem. To find and track the fingers we use blob analysis on a coloured glove (see Fig.3).

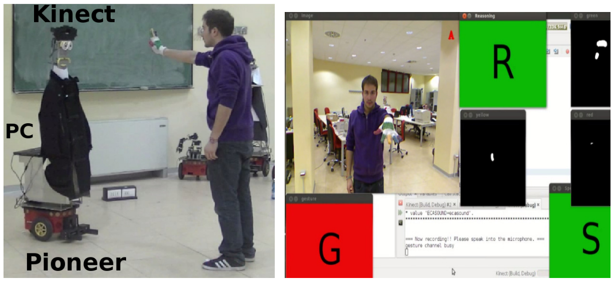

Fig. 3. Multimodal interaction with a mobile platform: a picture of an interaction and a screen-shot from the developed multimodal tool.

c) Gesture recognition: Our gesture classification approach is based on Hidden Markov Models (HMM) which is one of the most common choices in literature. We introduced the following set of labels for gestures: *Point at*; *Take*; *Drop*; *No*; *Give*; *Come here*; *Search*; *Stop*. We introduced the following features: (a) The 3D coordinates of the shoulder, elbow, and hand joints; (b) The 3D angles between the shoulder and the elbow, and the 3D angles between the elbow and hand; (c) An integer value representing the hand pose (e.g., open, close, pointing); (d) A boolean value, that indicates if the hand is directed toward the camera with the palm or with the back. Information extracted from Kinect is subject to noise that can complicate the gesture classification problem. To cope with this problem we deployed a Kalman Filter that estimates the 9 parameters corresponding to the 3D points of the user's hand, elbow and shoulder. Our points are specified as angles in the polar coordinate space, normalized in the interval $[0, 8]$ to reduce noise, as in [17], this allow us to satisfy robustness requirements. We collected 16 samples of each of the 8 gestures from a population of 10 subjects (5 males and 5 females). Given the gesture corpus, we defined the HMMs prototypes associated with the gestures. We introduced the following setting: vector size 17 for the features; 3 hidden states associated with the gesture initial, middle end ending phases; 3 Gaussian mixtures on each state and transition matrix representing the probability of transition between two states. Then, we introduced and trained 8 HMMs, one for each gesture using the corpus. To implement the HMMs we used the HTK (Hidden Markov Model ToolKit) [18]. HTK is primarily used for speech recognition tasks, but it has also been adapted to other applications, like gesture recognition. The tools provide support for speech analysis, HMM training, testing, and results analysis. The decoder of the HMM provides the N-Best list composed of the three best scores from the HMMs.

d) Speech Recognition: Data from microphone is converted into text strings by an Automatic Speech Recognition (ASR) unit and then analyzed by a Spoken Language Understanding (SLU) unit to extract meaningful information. To collect audio we analyze small windows of data, each one of size $SR/6$, where SR is the sampling rate, set to 44.1 Hz. We calculate the max amplitude A for each window and compare it to a threshold δ_s (we detect a spoken segment if $A > \delta_s$). Google speech tools are used for the speech-to-text conversion. As for SLU, we exploit a frame-based method. A frame is a memory structure composed of a set of slots. Some slots could be filled with other frames, creating tree-like structures. After the ASR has produced a text string corresponding to the spoken message, the SLU can parse this sentence to build a list of hypothesized frames. In our context, we define three frames: *SimpleAction* (commands like "No" that do not need additional data), *DirectedAction* (commands like "Take" that may need additional data), and *Object* (represents the target of a DirectedAction).

e) Fusion Engine: The *Action Classifier* is implemented by a SVM. Our SVM uses a feature vector of 7 parameters. The first 6 parameters represent the 3 best gestures obtained from the HMMs, with their associated scores, while the 7-th parameter is the value of the slot "name" of the frame produced by the SLU unit, representing the spoken command of the user. The SVM is implemented using LibSVM [19]. We created our model as a C-SVC with a radial basis function kernel. To choose the appropriate values for the γ and C parameters we used the python grid tool, provided by LibSVM. This tool uses a grid-search approach, where various pairs of (C, γ) values are tried and the one with the best cross validation accuracy is picked. We trained our model with $\gamma = 0.03125$ and $C = 128$. The *Object Classifier* is not implemented in this first prototype, because 3D coordinates are only specified by gesture commands and labels are only specified by speech commands. We created this unit for future development, where there could be more modalities that provide spatial information. This holds also for the *Modifier classifier* that is left as a future work.

f) Frame Builder: The Frame Builder groups information provided by the action, object and modifier classifiers. Information is represented as frames, extending the approach used in the frame-based SLU. We defined four different frames: Action (user commands), Object (target of the action), Point (where the action should be performed), and Modifier (attributes of the action). Frames are represented as Extensible Markup Language (XML) files.

g) Sample run: A user wants to command the robot to take the object on his right. He makes a pointing gesture while saying "take the object". Visual features are extracted from Kinect and analyzed by HTK, which produces the N-Best List $N_1 = [P(Point\,at) = 0.75, P(Drop) = 0.15, P(Take) = 0.1]$. Meanwhile the ASR analyses the spoken segment to produce the text string "take the object" with confidence $P(W) = 0.99$. The SLU unit analyses this string and produces the frame $F_1 = \{name = take; object = \{name = object; info = \text{``''}, location = \text{``''}\}\}$. The spoken and gesture components of the command are performed simultaneously by the user, so the Time Manager joins the responses from the recognizers in feature vector $v_2 = [Point\,at, 0.75, Drop, 0.15, Take, 0.1, Take]$. This vector is analyzed by the SVM, that produces the N-Best List $N_2 = [P(Take) = 0.97, P(Come\,Here) = 0.02, P(Give) = 0.01]$. The Frame Builder builds three different frames starting from N_2, F_1, and the analysis of the pointing gesture. The frame with the higher likelihood produced by the Frame Builder is $F_2 = \{name = \text{“}Take\text{”}, object = \{name = \text{“}object\text{”}, probability = \text{“0.99”}\}, point = \{x = 500, y = 300, z = 50\}, modifier = \text{“”}\}$.

A. System Testing

In order to evaluate the system performance, we performed an off-line testing for the classifiers and the multimodal architecture, using the recorded data. First, we tested our gesture classifiers and then our multimodal classifier. To evaluate the HMM used for gesture classification we performed the 10-fold cross validation on our data set. The model achieves an accuracy of 0.65. The test results are depicted in Tab.I. Looking at the confusion matrix we can see that some of the gestures, like *Search* and *No*, or *Take* and *Drop* are often misclassified. This was expected because we specifically defined the gestures to be ambiguous to test the fusion engine disambiguation ability. To evaluate the SVM model used for the multimodal fusion, we performed the 10-fold cross validation. Training data were obtained by testing HMMs with our gesture data set and collecting the responses. These responses were chained with different speech labels to obtain our desired data samples. We added a “No Command” label for monomodal commands. Samples containing only speech labels have been added to account for interactions with only speech. Our final dataset contains 1000 samples. The model generalizes well to independent data, achieving an accuracy of 0.97. The test results are shown in Tab.II.

For the on-line testing we selected 20 users (10 males and 10 females) and asked them to interact with a mobile robot (Pioneer 3DX) endowed with an on-board Kinect camera, a microphone, and a Laptop (see Fig.3). In this scenario, the users had to interact with the system trying to accomplish the following task: two colored objects had to be placed in specific locations while a hidden object had to be found by the robot. The robot movements were shown on a screen in order to provide a feedback to the user. Each user performed three different tests: in the first test they had to interact using only gestures, in the second only speech, and for the last test multimodal commands were allowed.

The quality of the interaction was assessed by asking the subjects to fill a specific HRI questionnaire, after each of the tests. The aim of this questionnaire, inspired by the HRI questionnaire adopted in [20], is to evaluate the naturalness of the interaction from the operator’s point of view.

The questionnaire is composed of a personal information section, containing the personal data and the experience with robotics, and a general feelings section, containing questions used to assess the perceived intuitiveness of our approach. In order to measure the level of confidence of the human with respect to the interaction, we asked about its naturalness (how did you feel about the naturalness of the interaction?), and about the legibility with respect to the robot point of view - e.g., if the robot understands the human intention expressed through the interaction - (Did the robot react accordingly with your behavior?). Scores were in the interval $[1 - bad, \ldots, 5 - excellent]$. Results are shown in Tab.III. Looking at the collected results we can see that, as expected, multimodal interaction perform better than both speech and gesture-based interactions. Most of the users found multimodal commands more natural and efficient to use. Our gestures were considered to be fairly natural and users needed only a short training phase to learn how to use them to interact with the system. Speech commands were considered very efficient and, on average, more natural than gestures. Our gesture set is composed of a sequence of codified movements that, while designed to be natural and easy to use, are not as rich as natural language, which allows users to interact with the system without having to remember our gesture set and how to execute them. The exception to this rule were pointing gestures, which were considered by some users to be more natural than speech to give the robot the location of the item to take. The results table contains the time that users took to complete the test. As expected gestures test durations were longer than speech or multimodal tests. Multimodal tests were shorter because users gave both location and action commands to the robot using gesture and speech simultaneously.

TABLE III

PERFORMANCE EVALUATION AND QUALITATIVE ANALYSIS.

Modality	Average Score	Accuracy	Average Time (min)

Gesture	naturalness = 3.8
	legibility = 3.4
		56%	3.1±1.3

Speech	naturalness = 4.0
	legibility = 4.2
		83%	2.3±0.5

Multimodal	naturalness = 4.8
	legibility = 4.4
		91%	2.1±0.5

V. CONCLUSIONS

We presented a multimodal framework for natural, robust, and flexible human-robot communication and interaction. The proposed system is intended to be extensible and easy to modify. Most of the multimodal HRI systems rely on a dominant modality, in contrast, in our system users are free to express their instructions as combinations of different modalities. For this purpose, we introduced a multi-layered late fusion approach, based on classification. In this work, we presented the system at work in a simple case study where a human operator is to accomplish pick-place-carry tasks interacting with a robot through gestures and speech. Using a fusion strategy based on SVM, an original approach in HRI multimodal systems, we achieved classification accuracy comparable with the state of art. The system was explicitly designed to allow for incremental developments, adding new input modalities or changing the classification strategies. New input channels, like user emotions, gaze detection, and full body movements can be added. Emotions, in particular, could be exploited to modulate the user commands providing a broader range of actions. Different commands, even with the accuracy resulting from multimodal fusion, can still be ambiguous and be interpreted in different ways in different

TABLE I

CONFUSION MATRIX FOR GESTURE RECOGNITION WITH PRECISION, RECALL AND F-MEASURE

	Point at	Come Here	Give	Search	Take	Drop	No	Stop	P	R	F
Point at	15.2	0.1	0	0	0.2	0.4	0	0.1	0.96	0.95	0.95
Come Here	0.3	7.3	0.4	1.4	3.1	2.5	0.4	0.6	0.48	0.45	0.47
Give	0	0.1	13.8	0	0.6	0.8	0	0.7	0.78	0.86	0.82
Search	0	2.3	0	9.2	0.1	0.1	4.1	0.2	0.70	0.57	0.63
Take	0.2	2.4	0.4	0	8.7	4.1	0	0.2	0.44	0.54	0.49
Drop	0	2	0.6	0	5.3	7.7	0	0.4	0.45	0.48	0.46
No	0.1	0.6	0.2	2.5	0.5	0.1	10.6	1.4	0.67	0.66	0.66
Stop	0	0.2	2.2	0	1	1.1	0.6	10.9	0.75	0.68	0.71
Average									0.65	0.65	0.65

TABLE II

CONFUSION MATRIX FOR MULTIMODAL RECOGNITION WITH PRECISION, RECALL AND F-MEASURE

	Point at	Come Here	Give	Search	Take	Drop	No	Stop	Go	P	R	F
Point at	7.8	0.1	0.1	0	0	0	0	0	0	1	0.97	0.98
Come Here	0	10.5	0.3	0	0.2	0	0	0	0	0.98	0.95	0.96
Give	0	0	12.8	0	0.2	0	0	0	0	0.87	0.98	0.92
Search	0	0	0.1	8.8	0.1	0	0	0	0	1	0.97	0.98
Take	0	0.1	0.8	0	22.1	0	0	0	0	0.96	0.96	0.96
Drop	0	0	0.5	0	0	10.5	0	0	0	1	0.95	0.96
No	0	0	0	0	0.2	0	7.6	0.2	0	0.98	0.95	0.96
Stop	0	0	0	0	0	0	0.1	12.9	0	0.98	0.99	0.98
Go there	0	0	0	0	0	0	0	0	4	1	1	1
Average										0.97	0.97	0.97

situations. Moreover, failures and ambiguities may arise in very noisy environment or simply due to the use of network dependent sensors. To solve this problem, in future work, we propose to introduce another layer, called *Dialogue Manager*, which interacts with the fusion engine to integrate the information about the human-robot dialogue context in interpretation of the user commands.

ACKNOWLEDGEMENT

This work has been supported by EU FP7 "SAPHARI" under grant agreement no. ICT-287513.

REFERENCES

[1] C. Heyer, "Human-robot interaction and future industrial robotics applications." in *IROS*. IEEE, 2010, pp. 4749–4754.

[2] A. D. Santis, B. Siciliano, A. Luca, and A. Bicchi, "An atlas of physical human-robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2007.

[3] A. Bannat, J. Gast, T. Rehrl, W. Rösel, G. Rigoll, and F. Wallhoff, "A multimodal human-robot-interaction scenario: Working together with an industrial robot," in *Proc. of Int.Conf on HCI Part II: Novel Interaction Methods and Techniques*. Springer, 2009, pp. 303–311.

[4] R. A. Bolt, "Put-that-there: Voice and gesture at the graphics interface," *SIGGRAPH Comput.Graph.*, vol. 14(3), pp. 262–270, 1980.

[5] H. Holzapfel, K. Nickel, and R. Stiefelhagen, "Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3d pointing gestures," in *Proc of ICMI*. ACM, 2004, pp. 175–182.

[6] B. Burger, I. Ferrané, F. Lerasle, and G. Infantes, "Two-handed gesture recognition and fusion with speech to command a robot," *Auton. Robots*, vol. 32, no. 2, pp. 129–147, 2012.

[7] M. Yang, S. Wang, and Y. Lin, "A multimodal fusion system for people detection and tracking," *Int. J. Imaging Syst. Technol.*, vol. 15, pp. 131–142, 2005.

[8] L. Wu, S. L. Oviatt, and P. R. Cohen, "From members to teams to committee-a robust approach to gestural and multimodal recognition," *Trans. Neur. Netw.*, vol. 13, no. 4, pp. 972–982, July 2002.

[9] A. P. Loh, F. Guan, and S. S. Ge, "Motion estimation using audio and video fusion," in *ICARCV*. IEEE, 2004, pp. 1569–1574.

[10] W. H. Adams, G. Iyengar, C.-Y. Lin, M. R. Naphade, C. Neti, H. J. Nock, and J. R. Smith, "Semantic indexing of multimedia content using visual, audio, and text cues," *EURASIP J. Adv. Sig. Proc.*, vol. 2003, no. 2, pp. 170–185, 2003.

[11] M. Gandetto, L. Marchesooti, S. Sciutto, D. Negroni, and C. S. Regazzoni, "From multi-sensor surveillance towards smart interactive spaces," in *Proc of ICME 2003 - Vol. 2*. IEEE, 2003, pp. 641–644.

[12] L. Kessous, G. Castellano, and G. Caridakis, "Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis," *Journal on Multimodal User Interfaces*, vol. 3, no. 1, pp. 33–48, 2010.

[13] J. Fiérrez-Aguilar, J. Ortega-Garcia, D. Garcia-Romero, and J. Gonzalez-Rodriguez, "A comparative evaluation of fusion strategies for multimodal biometric verification," in *AVBPA*, ser. LNCS, J. Kittler and M. S. Nixon, Eds., vol. 2688. Springer, 2003, pp. 830–837.

[14] J. A. Prado, C. Simplício, N. F. Lori, and J. Dias, "Visuo-auditory multimodal emotional structure to improve human-robot-interaction," *I. J. Social Robotics*, vol. 4, no. 1, pp. 29–51, 2012.

[15] J. A. Larson, T. V. Raman, D. Raggett, M. Bodell, M. Johnston, S. Kumar, S. Potter, and K. Waters, "W3C multimodal interaction framework," 2003. [Online]. Available: http://www.w3.org/TR/mmi-framework/

[16] M. Bodell, D. Dahl, I. Kliche, J. Larson, R. Tumuluri, M. Yudkowsky, M. Selvaraj, B. Porter, D. Raggett, T. Raman, and A. Wahbe, "Multimodal architectures and interfaces," 2011. [Online]. Available: http://www.w3.org/TR/mmi-arch/

[17] X. Wenkai and E.-J. Lee, "Continuous gesture trajectory recognition system based on computer vision," *Applied Mathematics and Information Sciences*, 2012.

[18] S. Young, G. Evermann, and et al., *The HTK Book*, 2006.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[20] M. Duguleana, F. G. Barbuceanu, and G. Mogan, "Evaluating human-robot interaction during a manipulation experiment conducted in immersive virtual reality," in *Proc. of ICVMR 2011: new trends-Vol. Part I*. Springer-Verlag, 2011, pp. 164–173.