Using body gestures and voice commands for assistive interaction

Răzvan Gabriel Boboc, Mihai Duguleană and Gheorghe Leonte Mogan

***Abstract*—**This paper discusses a human-machine interaction paradigm based on an operator's body gestures and voice commands for assistive applications. In the context of the humanoid robots already present on the market for the large public use, assistive robotics became a wide usage area to exploit the potential synergy of human-robot cooperation in order to extend and enable human activities that would otherwise be difficult or even not possible for the human alone. To enable these applications, simple and natural communication and interaction means are needed. The algorithms presented in this paper can be used for solving various assistive tasks and are based on Dynamic Time Warping (DTW) and Isolated Word Recognition (IWR). The system is tested successfully on the particular case of NAO humanoid robot, within an experimental scenario.

***Keywords*—**gesture recognition, voice commands, DTW, assistant robot, HRI.

I. Introduction

THE recent research in human-robot interaction is focused on creating domestic applications, with an increasing number of personal service robots that invade our homes or offices. Intelligent robots provide their support in many unpleasant, tedious human activities. These robots need to be capable of acquiring sufficient understanding of the environment, being aware of different situations, detecting and tracking people, as well as establishing a successful communication with humans in order to be able to cooperate with them [1].

An assistant robot should be able to interpret the verbally or non-verbally given instructions of the human [2]. In such context, researchers strive to find new simpler, more intuitive and human-like ways of interaction, that at the same time require less computational power and less sophisticated sensor devices. Along with other more recent approaches, the use of body gestures still remains a natural and thus an attractive alternative to cumbersome interface devices for human-computer interaction (HCI). Among various actions, the pointing gesture is natural, and perhaps, the most intuitive interaction paradigm, effective even in noisy environments and useful for commanding or simply messaging a robot [3].

In this paper we focus on the development of natural human-robot communication by means of human speech and gesture commands. In particular, we focus on using Dynamic Time Warping (DTW) for gesture recognition. The resulting module is used in combination with voice recognition to create human-like capabilities and behavior of the assistant robot. Thanks to this approach the robot gathers a very powerful ability: that of moving in an indicated direction and perform a required task - a High Level Interaction (HLI) paradigm [4] that we refer hereinafter as "point-and-command". Basically, this interaction metaphor is about indicating the robot a spatial location and a task to be performed there.

II. Background

Robots have been used as research tools in a variety of applications [1], [5], [6]. Some of them focus on how robots are accepted in the current society [7], suggesting an increasing presence of intelligent robots in our daily life, provided natural interaction is enabled. Latest research points the use of gestures as a way of interacting with computers and robots, as a natural and intuitive way of communication or option selection [1], [8].

There are many techniques used for gesture recognition [9], [10]. Commonly, these techniques are divided in two main categories: sensor-based and vision-based. While for the first category, the user is forced to bear different sensing devices attached to his body (gloves, magnetic trackers), in the vision-based approach the user does not require to wear any contact devices. The technique uses a set of visual sensors and algorithms to recognize gestures [8]. At the same time, gestures can be static or dynamic. For detecting dynamic gesture recognition in real time, there are issues in determining the start and the end points of a meaningful gesture pattern from a continuous stream [11]. While static gesture (pose) recognition can typically be accomplished by template matching and pattern recognition techniques, the dynamic gesture recognition problem involves the use of more advanced techniques [12].

Given these observations, researchers have proposed various solutions to optimize recognition of gestures [3], [11]. In this paper we will refer only to those based on vision,

This paper was sopported in part by the NAVIEYES research project (project code PN-II-PT-PCCA-2013-4-2023), financed by UEFISCDI.This work was supported in part by the Sectoral Operational Programme Human Resources Development (SOP HRD), ID134378, financed by the European Social Fund and the Romanian Government.

R. G. Boboc is with the Automotive and Transport Engineering, Transilvania Universtity of Braşov, 500024 Romania (phone: +40 268 236537; e-mail: razvan.boboc@ unitbv.ro).

M. Duguleană is with the Automotive and Transport Engineering, Transilvania Universtity of Braşov, 500024 Romania (e-mail: mihai.dugulean@ unitbv.ro).

G. L. Mogan is with the Automotive and Transport Engineering, Transilvania Universtity of Braşov, 500024 Romania (e-mail: mogan@ unitbv.ro).

generally body gestures. As shown in [13], the most widely used techniques for recognizing body movements are Hidden Markov Model (HMM), Dynamic Time Warping (DTW), Finite State Machine (FSM) and Neural Networks (NN). HMM was used in [3] for recognizing pointing gestures in order to control a mobile robot. They used 3D particle filters and a cascade of two HMM to estimate the pointing direction, dealing both with large and small pointing gestures. In [14] a probabilistic model, dynamic Bayesian network (DBN) was used for hand gesture recognition, which includes HMMs and Kalman filters. Also, NN in combination with HMM was used in [15] for hand gesture recognition, but the algorithm involves high computational costs.

Dynamic Time Warping was first used for speech recognition [16], but was extended also to other areas, including gesture recognition [17]. As we have seen above, there are several techniques used for detection and recognition of human gestures, but the most popular are HMM and DTW. Some papers have demonstrated that better results can be obtained with DTW instead of HMM both in voice recognition (animal vocalization) [18] and gesture recognition [12].

A. Gesture recognition with DTW

In order to detect gestures with a video camera, pattern matching technique or other similar algorithms can be used. Pattern matching involves the use of recorded drawings of gestures that serve as templates against which detected gestures can be compared. An example of such a technique is DTW, a template matching algorithm. The patterns are in this case a time sequence of measurements. DTW computes the cumulative distance between each pair of values of both time sequences, giving a measure of similarity between the two time sequences.

Various improvements have been made to the DTW algorithm, to make it more efficient, according to various authors. The methods used to make DTW faster fall into three categories [19]: constraints, data abstraction, indexing. In [20] a parallelisation of the original DTW algorithm is presented, in order to monitor multiple data streams using graphic processor units (GPUs). Lately, a probabilistic approach was proposed in [21]. Our technique combines data abstraction with lower bounding technique to improve performance.

B. Voice commands recognition

There are many studies on speech recognition with specific interest in commanding robots. The main goal of almost any work in this area is to achieve a natural-language communication with the robot assistant.

Various algorithms are used to achieve speech recognition. One of them is the Dynamic Time Warping (DTW), which is based on pattern comparison, fairly similar to the one used in video processing [22]. Other studies use Hidden Markov Models (HMMs) [23], empowering statistics to handle a specific vocabulary. Artificial Neural Networks (ANNs) is another technique used one its own or combined i.e. with HMM for achieving speech recognition [24].

For this study, we use a vocabulary approach based on Microsoft Kinect speech recognition library. The algorithm behind the library is as follows: an audio stream taken from Kinect sensor is parsed and then vocal utterances are interpreted. If the engine recognizes some elements, they are sent to the processing unit. If the command is not recognized, it removes that part from audio stream.

III. Overview of the Proposed System

To identify the human gestures we used a Kinect camera mounted in front of the user. This corresponds to a human sitting at his desk situation (Fig. 1). The Kinect camera records the movements and listens for voice signals recognizing the words spoken by the user. As a result, it sends a 'wake up' command to the robot.

The assistive robot is physically able to autonomously walk to a specified location, recognize an object and grab it. In order to be able to command the robot for performing these tasks, a vocabulary of words and gestures was designed. It consists of several arm movements and speech commands which may be combined in several ways. Since the environment can be noisy or with poor lighting conditions, some commands have been chosen for use in both modes of interaction (by gesture and voice). Thus, for starting the interaction with the robot, the users can perform an initialization gesture or can speak the robot name.

The robot has a fixed initial position, which is marked with a NAOmark, as in Fig. 2. Objects are placed in different positions in the room. After the connection with the robot is established, the user can ask the robot to bring him an object indicated by pointing gesture (Fig. 8). The robot will move in the indicated direction, will identify the object and will grab it, then will move back to the user. If the robot encounters certain

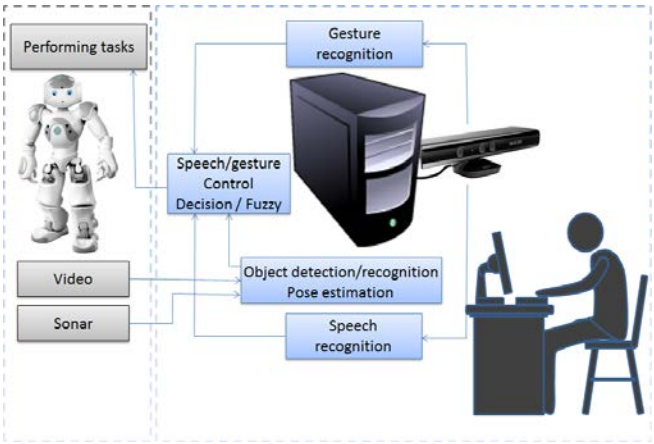

Fig. 1. Overall view of the system architecture

difficulties such as obstacles or can't identify an object, it asks by voice or by a predefined gesture. In [4] this paradigm (name here "point-and-command") was defined as a high-level command, which do not explicitly specify the target location, but help robots in autonomous target selection.

A. Hardware and software prerequisites for theoretical and experimental studies

Microsoft Kinect sensor was used in this work for both gesture and speech recognition. This sensor is a low cost capture device originally developed for the Xbox 360 video game console. It contains a RGB-D camera for image acquisition and an array of four microphones for capturing

sound and locating its source. Due to its benefits, Kinect was used for research purposes, enabling touch-less interactions through voice and gesture. Users can move freely, without being constrained to wear other sensors or devices on their body.

Kinect for Windows SDK was used, which is a toolkit that provides an interface to interact with the device. It provides API libraries for .NET and C/C++ applications that run on Windows platforms.

Kinect SDK tracks 3D coordinates of 20 body joints in real time (30 frames per second) and the obtained joint positions are used to recognize the gesture or posture which will command the robot.

A desktop PC is the main processing unit. As is illustrated

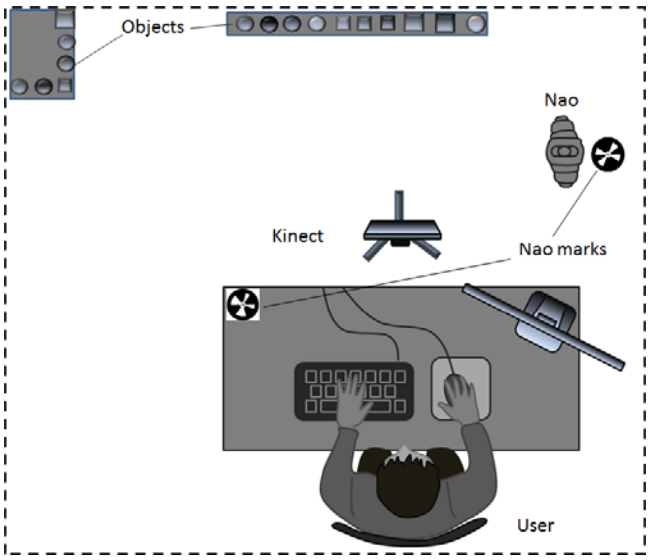

Fig. 2. Layout of the testing room

in Fig. 1, on this computer runs the application that allows gesture recognition, voice recognition, a speech/gesture integrator system and the communication with the robot. The computer is equipped with an Intel core i7 X 990 CPU 3.47 GHz, 12 GBs RAM.

IV. Metods

A. Human's gestures recognition

In gesture recognition, a comparison between two sequences is essential. Dynamic gesture recognition typically contains two components: segmentation and recognition [25]. Segmentation is the process of locating a gesture from a frame sequence. We use DTW technique to assess the similarity between two video sequences obtained from Kinect sensor. The input data is compared with a predefined sequence; the two sequences are aligned in order to determine the minimum cost path. This minimum cost represents the optimal alignment between the two sequences, which means that the corresponding gesture is considered to be recognized.

A problem that occurs is to know when to start the gesture recognition procedure, because if a gesture differs only in starting position from the predefined sequence, the result of the alignment with DTW technique will be very different. For this, we choose to use an initialization phase, which consists of a simple word spoken by the user, by which the robot is warned that the user wants to start a gesture interaction. When that predefined word is pronounced, the program automatically starts the gesture recognition process. The end of the gesture is considered when the hands stops moving.

B. Structure of the proposed algorithm

The flow diagram of the proposed algorithm is shown in Fig. 3. As it can be seen, the first stage is to detect the human. After that, features are extracted. The DTW algorithm is applied to the extracted vectors and if the gesture is recognized, then the robot will perform the requested action. Otherwise, it will initiate a speech interaction in order to ask for further details.

The proposed gesture recognition algorithm involves 4 steps: 1) automatic human detection, 2) feature extraction, 3) a gesture pattern stage, where gestures are compared with reference gestures, 4) gesture recognition (Fig. 3).

The first stage of the algorithm is to detect the human body. This is facilitated by the Kinect sensor that can find the skeleton using a very fast and accurate recognition system that requires no setup, because a learning machine has already been instructed to recognize the skeleton. Joint positions are obtained like in Fig. 4. For this study, just the arm joints are relevant, especially hand, wrist and elbow joints. The coordinates of that joints form a feature vector.

For simplicity, two assumptions were made: first, it was assumed that a single person is presented at a time in front of the sensor and second, that person has a sitting posture. The initiative of initiating an interaction with the robot belongs to operator. After the initialization stage, the system is 'prepared' to recognize the gesture performed by the user. The gesture should be performed quickly because is represented on 33 frames. After the 33th frame, the feature vector is compared with sample gestures. Once a gesture is recognized, depending on its significance, the system will decide what task the robot should be performed.

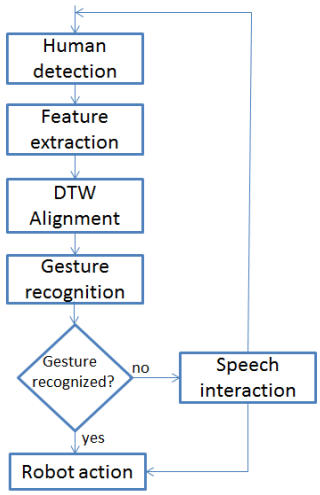

Fig. 3. Gesture recognition flow diagram

Feature extraction. The most important information about a body gesture is the motion of limbs. In this case, upper limbs are relevant because the system was designed for humans sitting on chair. The motion of an arm is described by its trajectory in space. This trajectory represents a time sequence of positions of the arm.

The feature vector captured from Kinect contains the x, y positions of each arm joint. This vector is then preprocessed in order to prepare it for DTW computation. Preprocessing stage includes eliminating missing or redundant data and other variations and set vector length. The feature vectors that are

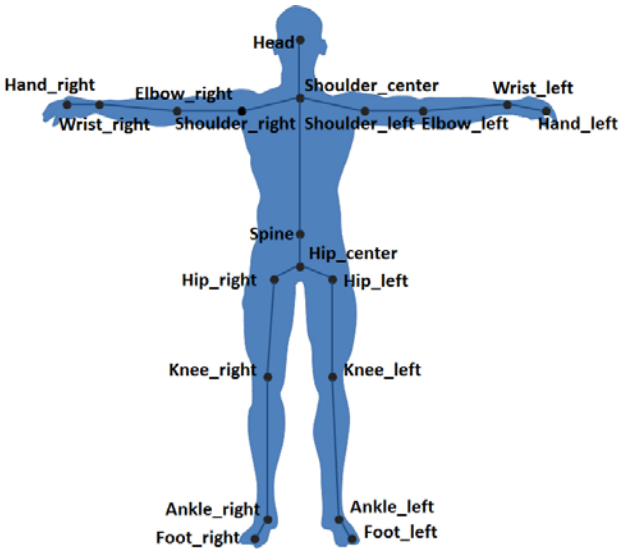

Fig. 4. Kinect joints

characteristic for a command gesture are extracted and then stored into a database.

The minimum distance from the Kinect device for an accurate detection is 60 cm. The sensor is placed on a tripod, in front of the user's desk (see Fig. 8).

C. DTW Method

Dynamic time warping (DTW) is a powerful technique in the time-series similarity search [26]. An overview of this method is given below.

Given two time series sequences: $x = x_1, x_2, \dots, x_i, \dots, x_n$ of length n and $y = y_1, y_2, \dots, y_j, \dots, y_m$ of length m, a n-by-m matrix can be obtained, where each element of the matrix represents the distance between two elements of the time series, named *cost matrix*. The optimal alignment between x and y needs to be found, such that the overall cost is minimal. A *warping path* $w = w_1, w_2, \dots, w_k, \dots, w_p$ defines such mapping between the elements of the two time series (Fig. 5).

DTW(x, y) = min $\sum_{k=1}^{p} d(w_k)$

The DTW warping path is constrained to follow some restrictions, like monotonicity, continuity, warping window, slope constraint and boundary conditions [27].

The cost for the optimal alignment is recursively obtained by:

$$\gamma(i,j) = d(x_i, y_j) + \min[\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)]$$

As we said above, new versions of DTW algorithm were developed for improving speed, while others were developed for improving accuracy. The lower bound technique for DTW was first proposed in [28]. A lower-bound function for DTW is a function that always returns a value smaller than or equal to the actual DTW distance. The most cited lower bound is LB_Keogh, which uses the warping path to compute an envelope on the warping cost. Improved versions of the envelope technique were proposed in [29].

The warping envelope of time series x is represented by the pair U(x) and L(x), where:

$U(x)_i = \max_k \{x_k \mid |k-i| \le \gamma\}$

$L(x)_i = \min_k \{x_k \mid |k-i| \le \gamma\}$, i = 1, .., n, where γ is a local constraint

The lower bounding function LB_Keogh is defined as:

$$LB\_Keogh(x,y) = \sqrt{\sum_{i=1}^{n} \begin{cases} (y_i - U_i)^2, y_i > U_i \\ (y_i - L_i)^2, y_i < L_i \\ 0, \quad otherwise \end{cases}}$$

In order to satisfy the requirements of a robust gesture recognition system for interaction with a mobile robot, we propose an improved version of DTW, that combines several techniques, as will be shown below.

The time complexity of DTW algorithm is O (n*m) for two sequences like those presented above, which makes the method not practice for longer time series. Although our sequences, represented by feature vector are small, we decide to use the algorithm presented in [19], which is O(n) both in time and space. The presented method, named FastDTW, uses a multilevel approach with three steps: coarsening, projection and refinement. First, the size of time series is reduced by averaging adjacent pairs of points, and then a warp path is calculated for this lower resolution, which will be used to find the warping path for higher resolutions. Finally, the warping path is refined, searching for the optimal path on each side of the projected path, according to a *radius* parameter, that indicates the number of cells to be evaluated.

FastDTW was slightly modified. After the coarsening step, the minimum distance warping path was obtained using another technique, a lower bound function introduced in [29], that that offers a plausible speedup [30]. Given the time series presented in section 2.1, LB_Improved is defined as:

LB_Improved (x,y) = LB_Keogh(x,y) + LB_Keogh(y, H(x,y)) , where H(x,y) is the projection of x on y:

$$H(x,y)_i = \begin{cases} U(y)_i, & x_i \ge U(y)_i \\ L(y)_i, & x_i \le L(y)_i \\ x_i, & otherwise \end{cases}, \text{ i = 1, 2,…,n}$$

DTW compares the sequence obtained for an unknown gesture to one or more reference templates. Having more reference templates, the recognition rate will be higher, but the computing time also increases. For this reason, an approach implemented in [31] for speech recognition is used. This algorithm, named Quantized DTW, stores one reference model for each gesture. This algorithm was adapted for gesture recognition.

The Quantized DTW together with FastDTW and LB_Improved were combined in order to obtain a fast and accurate gesture recognition algorithm.

V. Human-Robot Interaction

HRI inputs are diverse, but we focus in this paper on vision and speech. Computer vision was used to process human gestures and to detect objects, while speech was used to exchange information between human and robot. User can give instruction to the robot using both gesture and voice, in the same way as people communicate with each other.

A. Gesture interaction

A gesture is a bodily movement made intentionally by a human in conversation, in order to aid in better understanding

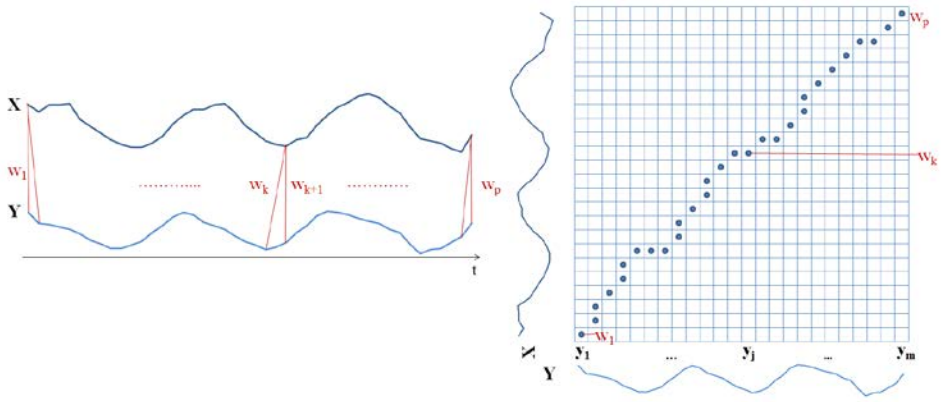

Fig. 5. a) The alignment of two time series (x, y) by DTW; b) and the mapping between them – the warping path (with blue dots)

of what he said. In human communication, hand, head and arm gestures play an important role.

In HRI domain, since assistive robots interact with non-expert users, natural interfaces are essential and therefore robots should be able to understand the modalities used by humans during interaction. The same as in human-human interaction, a gesture can provide information or to communicate intention to robot. A set of gestures was created, which represents the 'command vocabulary' for HRI. In Table 1 are shown the defined gestures. Most of the gesture were inspired from [32].

Special attention was given to pointing gesture because is an easier way to draw robot's attention indicating an object or a location in space and is useful for non-expert users. Once the gesture was been detected, the next step is to estimate the pointing direction. For this work, we need to calculate the angle between user's arm and shoulder center. Three joints from the skeleton describe this gesture: shoulder center, shoulder and hand (Fig. 6). The estimated angle was calculated using the following formula:

$$\alpha = acos\ \frac{v1.x \cdot v2.x + v1.y \cdot v2.y}{\sqrt{v1.x^2 + v1.y^2} \cdot \sqrt{v2.x^2 + v2.y^2}} * \frac{180}{\pi}$$, where v1, v2 are two vectors:

v1 = Shoulder – Hand

v2 = Shoulder_Center - Shoulder

The pointing gesture is used only when the robot is in the home position, knowing its orientation and the distance from Kinect. Otherwise, it does not know in which direction to go.

All gestures are made with arms, excepting two: head nod and head shake. We choose to use these gestures because are the most commonly used in interpersonal communication when they accept or reject something. As simple head tracking algorithm was used, taking into account the head rotation on the sagittal or transverse plane.

Some gestures have different meanings depending on the context. For example, when the user says 'Rotate left' and robot is moving, it will change direction of walking to left with 15°. If the robot is not moving, the same command will refer to robot' camera, and then it will rotate the head 15° to left.

To make the interaction more realistic some basic behavior for humanoid robot were developed (like shrugging, confused - *robot scratches its head*.

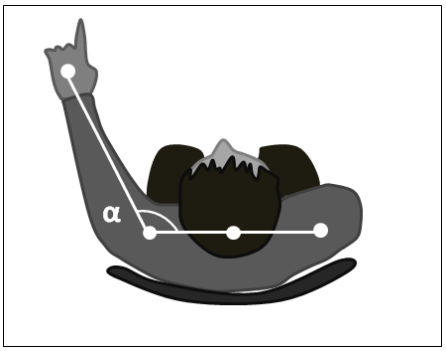

Fig. 6. Pointing angle

B. Speech interaction

As for gesture interaction, a set of speech commands was created. In Table 2 are shown the basic verbal phrases used in interaction, but is not a complete table because some of them can be combined to form predefined utterances, as it will be shown in below.

For starting the interaction with the robot, the user is required to say the robot's name ("NAO") in order to know that user is speaking with it.

The voice command system was created using Kinect for

Table I. Gesture vocabulary

Gesture name	Abb	Description	Meaning
Attention	A	One hand pointing up	'Hey!'
Big	B	Both hands are held at head level with large distance between them	'A bigger object'
Break	Br	One hand placed perpendicular to the other hand	'Time out!'
Calm	Ca	Both hands pressing down repetitively	'Go slowly'
Circle	O	Draw a circle in space	'An object like this'
Come	C	Hand moves repeatedly from outward toward the body	'Come here!'
Despair	D	Both hands are raised at head level	'What have you done?'
Doubt Shrug	Do	Hands are opened in an outward arc	'I don't know'
Head nod	HN	Head is tilt vertically once or several times	Acceptance
Head shake	HS	Head is turned left and right repeatedly	Rejection
Left	L	Left arm raised at shoulder level in the left side of the body	'Go left!'
Object	Ob	Hand points towards an object	'That object!'
Rectangle	R	Draw a rectangle in space	'An object like this'
Refuse	Re	One hand is moved outward in a wiping motion	Negation
Right	R	Right arm raised at shoulder level in the right side of the body	'Go right!'
Small	S	Both hands are held at head level with small distance between them	'A smaller object'
Space	Sp	Hand points into the space	'Go there!'
Sway	Sw	Both hands alternate in an up-down movement	'Keep going!'
To-Fro	TF	Both hands move from one side to the other	'Move there'
Triangle	T	Draw a triangle in space	'An object like this'
Turn left	TL	Both hands imitating the rotation of an object in counterclockwise direction	'Rotate left 15°!'
Turn right	TR	Both hands imitating the rotation of an object in clockwise direction	'Rotate right 15°!'
Wave	W	The at hand is outstretched, upward, with small sideways movements	Calls robot attention
Wipe	Wi	Both hands start near each other and move apart in a straight motion	Termination, finish
X	X	Hand crossed	Exit application

ISBN: 978-1-61804-325-2

Table II. Speech vocabulary

Command	Abb	Meaning
Nao!	N	Start interaction
Stand up!	SU	Robot stand up from rest position
Sit down!	SD	Robot sits in the rest position
Go there!	GT	Robot goes in the indicated direction
Go left!	GL	Go in the left direction, rotating with 90°
Go right!	GR	Go in the right direction, rotating with 90°
Turn left!	TL	Turn left 15°
Turn right!	TR	Turn right 15°
Grab the object!	GO	Robot will autonomously catch an object, calculating the distance to it
Drop object!	DO	Robot release object from his hand
Leave object!	LO	Robot raise its arm and open the hand
Let me control you!	C	Teleoperation mode
Learn this!	LT	Learn a new task
Bring it to me!	B	Bring object to user
Open hand!	OH	Robot open its hand
Close hand!	CH	Robot close it hand
Thank you!	TK	Robot wait for another command
Yes!	Y	Acceptance
No!	N	Rejection
Stop!	S	Robot stops the action it perform
Exit!	E	Close the application

Windows SDK, combined with Microsoft Speech Recognition (MSR) API. Kinect SDK provides various audio capabilities and Microsoft Speech platform provides classes to work with speech recognition captured by Kinect sensor, converting spoken words to written text. The sensor can detect audio that is within ± 50 degrees in front of sensor and also supports up to 20 dB of ambient noise cancellation.

Microsoft Speech Recognition has advanced grammar and vocabulary and it doesn't require any training for the models. The user should create his grammar with the desired keywords. A Kinect handler will initialize audio stream and will start the audio capturing. Once the speech recognition engine starts, user will load the grammar and from now the system is ready to listen from Kinect. Then, each recognized word has a confidence level, showing the reliability of the detection.

C. *Gesture/speech fusion*

Speech and gesture recognition modules are run simultaneously. After the command “Attention” (by gesture) or “NAO” (by voice), the system waits for another command that can be by gesture or by voice. There are four possibilities resulting from combination of interaction modalities: only gesture (G), gesture+voice (GV), voice+gesture (VG), only voice (V). Each command is sent to the decision system, which is based on different rules and, according to these rules, the task that have to be performed is identified. If the commands are GV or VG, the system decide if gesture command is congruent or not with voice command. The tasks or actions implemented are the following: navigation (N), fetching (F), grabbing (G), pushing (P), and teleoperation (T).

The following rules constitute part of the knowledge base and express how the system has to react:

If <gesture command> is C and <voice command> is GL then task is N

If <gesture command> is Sp and <voice command> is B then task is F

If <gesture command> is Ob and <voice command> is GO then task is G

If <gesture command> is A and <voice command> is C then task is T

If a gesture command is incongruent with the voice command, the robot will respond by predefined behaviors or by speech. Otherwise, the system decides the action given by one command only or both congruent commands.

D. *Robot tasks*

Programming of the robot consists of path planning according to the target. So a *task* in our work is defined as movement to a location plus a simple manipulation (two sub-programs). Each task has so need 2 essential inputs: location and handling. These two information are obtained by the robot through dialogue: the robot asks by voice and human answer by one of the mentioned metaphors. We choose to use some simple tasks that are commonly found in home environments: push, fetch.

We choose also to use only the basic capabilities of the robot and not to enhance them. The system uses an external computer to perform all the computations concerning gesture/speech interaction, video processing, and so on.

For grabbing an object task, an algorithm inspired from [33] was used for measuring the distance to the object with video camera and sonar sensors.

For simplicity, we choose objects with known shapes: balls, cubes, and cones (Fig. 7). Each object has some particular properties or attributes that are shown in Table 3.

Shape attribute refers to volumetric property of the object (2D form). The software associates the object name with a simplified representation of the object, corresponding to shape, color, and size properties.

An image taken by the robot’s camera is first segmented using a color detection algorithm using OpenCV. In this operation, the robot tries to separate the object in the scene from the background. The shape of the objects is detected using edge detection algorithm [34].

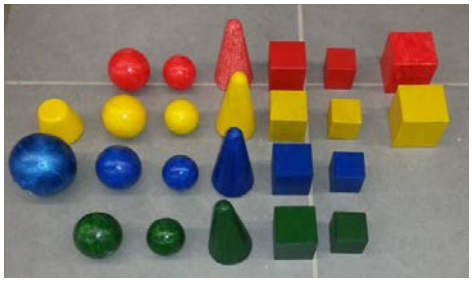

Fig. 7. The objects used for experiment

VI. Results

In this section will be presented the experiment conducted with the aim to test the performance of the system and to evaluate operation and precision dialogue in global application. The experiments were conducted in our institute environment. The user asked NAO to follow his instructions given by means of multimodal requests. NAO is asked to go in a desired direction indicated with pointing gesture. The robot will navigate in that direction and will bring to the user an

object whose name and properties are sent to robot by voice command.

A simple dialogue between user (U) and NAO humanoid robot (N) is proposed. The experiment was conducted by 4 persons for 3 times. The user asks NAO to bring him a red ball located in a certain position in the environment. Below is shown the whole dialogue.

U: 'NAO!'
N: 'Yes, I hear you"
U: 'Please, give me the red ball from there!'
N: 'Can you show me how the red color looks like?'
(*user show it a sample painted in red*)
N: 'What about the shape of the object?'
U: 'The ball has this shape'(user show it a circle drawn on a paper or by gesture – draw a circle in the air with his hand)
(*the robot walk in that direction – when it identify the red color, it will goes toward to identify the shape*)
N: 'Is that the object?'
U: 'No, I need a bigger one'
(*the robot will continue looking until it find a bigger ball*)
U: 'Grab the object!'
(*the robot decide if it can grab the object with one hand or with both hands*)
U: 'Bring it to me!'
(*NAO is looking for NAO mark and go in that direction*)
U: 'Leave it'
(*NAO leave the object*)
U: 'Thank you!'
N: 'Do you want another thing?'
U 'No'
(*NAO will go to home position*)

During the experiment more gestures have been used in order to test the performance of recognition algorithm, especially for navigation task. The confusion matrix among gesture commands for 4 users is shown in Table 4.

VII. Discussion

In this work we describe a framework for a natural and easy human-robot communication and interaction. While most of the multimodal HRI systems proposed in literature focus on a single modality, our system allows the users to express their instructions as combinations of gestures and speech inputs. The main strengths of our system are: the improved method of gesture detection, easy and natural interaction through gestures and voice commands, and the gestures and voice feedback provided by the robot.

Table III. Objects and their attributes

Object name	Attribute name	Values
Ball	Shape	Circle Triangle Square
Cube	Colour	Red Yellow Blue
Cone		White
	Size	Smaller Larger

The purpose of the interface is to allow expert and non-expert users to cooperate and interact with an assistant robot operating in a domestic environment. A gesture and a speech vocabulary were implemented and the commands can be sent by one or both modalities. So, a first objective of our research was to provide the robot with social interaction capabilities, which are essential for assistive robots applications.

Most of the work was focused on gesture interaction, specifically gesture recognition. An improved DTW method was implemented and tested, with good results both in accuracy and efficiency. The method increases the robot reactivity at the human requests enhancing the naturalness of the interaction. Combined with the speech/gesture capability resulted in a versatile interface that facilitates powerful interaction paradigms like the "point-and-commands" one.

However, there are some problems or limitations that were encountered during the experiments and that still need to be addressed. The recognition accuracy is dramatically affected when there are poor lightning conditions or noise in the operation environment. On the other hand, when multiple humans appear in the visual range of sensor, the system has difficulties in identifying the right user. In some situations the robot was unable to identify markers and lose orientation. Also, some smaller obstacles were not detected and sometimes the robot falls.

The above problems and others show that several further developments are needed to be addressed in our future research, as follows:

- Expanding the gesture vocabulary by adding also hand gestures, that are more intuitive and which can express more of the user wishes.
- Implementing a more advanced method to detect object with different shapes and colors and for object manipulation.
- Considering the possibility that more users wish to interact with the robot in the same time. In this case, the system must be intelligent enough to select the user that will interact with the robot.
- Developing more complex scenarios with a variety of tasks that have to be performed by the robot.

Table IV. Confusion matrix for navigation task

		Recognized gesture										
		B	Ca	C	L	R	Sp	Sw	TF	TL	TR	W
Performed gesture	B	90%	0	0	0	0	0	0	0	0	0	0
	Ca	0	100%	0	0	0	0	0	0	0	0	0
	C	0	0	90%	0	0	0	0	0	0	0	0
	L	0	0	0	95%	0	0	0	0	0	0	0
	R	0	0	0	0	90%	5%	0	0	0	0	0
	Sp	0	0	0	0	0	95%	0	0	0	0	0
	Sw	0	0	0	0	0	0	85%	0	0	0	0
	TF	0	0	0	0	0	0	0	90%	0	0	0
	TL	0	0	0	0	0	0	0	0	95%	0	0
	TR	0	0	0	0	0	0	0	0	0	95%	0
	W	0	0	0	0	0	0	0	0	0	0	90%

Fig. 8. The testing room. The user indicates the location by pointing and verbally the task to be executed

ACKNOWLEDGMENT

This paper was supported by the Project no. 240/2014, code PN-II-PT-PCCA-2013-4-2023, entitled "Intelligent car navigation assistant for mobile devices based on eye gaze tracking and head pose - NAVIEYES" financed by UEFISCDI and by the Sectoral Operational Programme Human Resources Development (SOP HRD), ID134378, financed by the European Social Fund and the Romanian Government.

REFERENCES

[1] V. Alvarez-Santos, R. Iglesias, X. M. Pardo, C. V. Regueiro, and A. Canedo-Rodriguez, "Gesture-based interaction with voice feedback for a tour-guide robot," *Journal of Visual Communication and Image Representation*, 2013.

[2] J. Richarz, A. Scheidig, C. Martin, and S. Mueler, "A Monocular Pointing Pose Estimator for Gestural Instruction of a Mobile Robot," *International Journal of Advanced Robotic Systems*, vol. 4, pp. 139-150, 2007.

[3] C.-B. Park and S.-W. Lee, "Real-time 3D pointing gesture recognition for mobile robots with cascade HMM and particle filter," *Image and Vision Computing*, vol. 29, pp. 51-63, 2011.

[4] A. Caltieri and F. Amigoni, "High-Level Commands in Human-Robot Interaction for Search and Rescue," in *The 17th annual RoboCup International Symposium*, Eindhoven, Netherlands, 2013.

[5] T. Breuer, G. Giorgana Macedo, R. Hartanto, N. Hochgeschwender, D. Holz, F. Hegger, *et al.*, "Johnny: An Autonomous Service Robot for Domestic Environments," *Journal of Intelligent & Robotic Systems*, vol. 66, pp. 245-272, 2012.

[6] M. Duguleana, F. G. Barbuceanu, A. Teirelbar, and G. Mogan, "Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning," *Robotics and Computer-Integrated Manufacturing*, vol. 28, pp. 132-146, 2012.

[7] M. M. A. de Graaf and S. Ben Allouch, "Exploring influencing variables for the acceptance of social robots," *Robotics and Autonomous Systems*, 2013.

[8] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A Kinect-based natural interface for quadrotor control," *Entertainment Computing*, vol. 4, pp. 179-186, 2013.

[9] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, pp. 311-324, 2007.

[10] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 677-695, 1997.

[11] J. Kang, K. Zhong, S. Qin, H. Wang, and D. Wright, "Instant 3D design concept generation and visualization by real-time hand gesture recognition," *Computers in Industry*, vol. 64, pp. 785-797, 2013.

[12] J. Carmona and J. Climent, "A Performance Evaluation of HMM and DTW for Gesture Recognition," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. vol. 7441, L. Alvarez, et al., Eds., ed: Springer Berlin Heidelberg, pp. 236-243, 2012.

[13] S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, pp. 1-54, 2012.

[14] H.-I. Suk, B.-K. Sin, and S.-W. Lee, "Hand gesture recognition based on dynamic Bayesian network framework," *Pattern Recognition*, vol. 43, pp. 3059-3072, 2010.

[15] C. Zhu and W. Sheng, "Online Hand Gesture Recognition Using Neural Network Based Segmentation," in *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, 2009.

[16] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *the 7th International Congress on Acoustics*, Budapest, Hungary, 1971.

[17] H. Li and M. Greenspan, "Model-based segmentation and recognition of dynamic gestures in continuous video streams," *Pattern Recognition*, vol. 44, pp. 1614-1628, 2011.

[18] J. A. Kogan and D. Margoliash, "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: a comparative study," *Journal of the Acoustical Society of America*, vol. 103, pp. 2185-2196, 1998.

[19] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, pp. 561-580, 2007.

[20] Jason Chang and M.-Y. Yeh, "Monitoring Multiple Streams with Dynamic Time Warping using Graphic Processors," in *Parallel Data Mining Workshop (IPDM)*, Mesa, Arizona, USA, pp. 11-20, 2011.

[21] Miguel Ángel Bautista, Antonio Hernández-Vela, Victor Ponce, Xavier Perez-Sala, Xavier Baró, Oriol Pujol, *et al.*, "Probability-Based Dynamic Time Warping for Gesture Recognition on RGB-D Data," *Advances in Depth Image Analysis and Applications*, vol. 7854, pp. 126-135, 2013.

[22] L. Hong and X. Li, "A selection method of speech vocabulary for human-robot speech interaction," in *IEEE International Conference on Systems Man and Cybernetics (SMC)*, pp. 2243-2248, 2010.

[23] S. O. Caballero Morales, G. B. Enríquez, and F. T. Romero, "Speech-Based Human and Service Robot Interaction: An Application for Mexican Dysarthric People," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1-14, 2013.

[24] P. Varchavskaia, P. Fitzpatrick, and C. Breazeal, "Characterizing and processing robot-directed speech," in *IEEE/RAS international conference on humanoid robots*, ed. Tokyo, Japan, 2001.

[25] J. Cheng, W. Bian, and D. Tao, "Locally regularized sliced inverse regression based 3D hand gesture recognition on a dance robot," *Information Sciences*, vol. 221, pp. 274-283, 2013.

[26] M. Zhou and M. H. Wong, "Boundary-based lower-bound functions for dynamic time warping and their indexing," *Inf. Sci.*, vol. 181, pp. 4175-4196, 2011.

[27] D. J. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," in *KDD Workshop*, Seattle, Washington, USA, pp. 359-370, 1994.

[28] B. K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Proceedingsof 14th International Conference on Data Engineering*, pp. 201-208, 1998.

[29] D. Lemire, "Faster retrieval with a two-pass dynamic-time-warping lower bound," *Pattern Recogn.*, vol. 42, pp. 2169-2180, 2009.

[30] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, pp. 275-309, 2013.

[31] T. Zaharia, S. Segarceanu, M. Cotescu, and A. Spataru, "Quantized Dynamic Time Warping (DTW) algorithm," in *the 8th International Conference on Communications (COMM)*, pp. 91-94, 2010.

[32] M. Kipp, "Gesture generation by imitation: From human behavior to computer character animation," Ph. D, Saarland University, 2004.

[33] G. Jingwei and M. Q. H. Meng, "Study on distance measurement for NAO humanoid robot," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 283-286, 2012.

[34] J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679-698, 1986.