# "Drive Me": a Interaction System Between Human and Robot

Stefan-Gheorghe Pentiuc
Faculty of Electrical Engineering and Computer Science
MintViz Lab, MANSiD Research Center
Stefan cel Mare University of Suceava, Romania
pentiuc@eed.usv.ro

Oana-Mihaela Vultur
Ştefan cel Mare University of Suceava
Suceava, Romania
vultur_oana@usv.ro

***Abstract*— This paper presents a new interaction system between human and robot, called "Drive Me", a system that allows users to "drive" a electric robot using natural and dynamic gestures.**

**The user stands in front of the depth camera and performs the gestures that practically move and "drive" the electric robot. The naturalness and the facility of performing the gestures let us know that "Drive Me" is a robust and easy to use interaction system between human and robot.**

**Gesture recognition has a high accuracy rate and that makes of "Drive Me" a system easy to use by different users, system that does not depend of the ambient light conditions.**

**We also made a performance evaluation of the "Drive Me" interaction system and we calculated the accuracy rate, the error rate, the precision, the recall, the sensitivity and the specificity of gesture recognition.**

***Keywords*— HRI (Human Robot Interaction); gesture recognition; robot; interaction system; computer vision; gesture interaction; system's evaluation***

## I. INTRODUCTION

Objective of the field of study Human-Robot Interaction (HRI) is the "understanding, design and evaluation of the robotic systems for use by/ with humans" [1]. HRI is one of the most challenging areas regarding gesture interaction.

The most application areas of HRI are: search and rescue, entertainment, military and police, assistive and educational robotics, space exploration, medical and health care, etc.

Most often people use gestures to highlight or explain the verbal message. They can give more expression to speech. In virtual reality applications gestures can also be used to navigate in the virtual environment [2] [3] [4], to use software applications on touch screens, to play games with smart phone, to interact with your computer [5], to simulate assembly operations [6], to control a device from real world like such as a surgical instrument, a industrial robot [7] or a mobile robot [8]. Cerlincă et al., for example, proposed in [7] a system that controls a industrial robot by 3D gestures performed with arms in a natural way. The algorithm used by the authors for gesture recognition is DTW (Dynamic Time Warping).

Interaction with robots will play an important role also in the future space missions. The astronauts during their exploring activities [9], or those at the Earth Control Station, will may communicate with the robot through speech or gesture interfaces in a as natural as possible manner.

A first step in recognizing gestures is to find a set of features with a great discriminatory power. In most cases, these features are calculated by the computer, and are very difficult to be understood by humans. In the paper [10] it is proposed a set of 17 measures to describe gestures by their spatial characteristics of the body movement, their kinematic performance, and the body posture. The paper also presents a body gesture analysis tool that automatically calculates these measures from a video stream.

In the control of a robot the arm movements and hand postures play an important role. In paper [11] there is presented a robust recognition system of hand gestures that uses a RGB-Depth sensor. To avoid noises and occlusions, Haar-like Steric features are used to represent the complex spatial relations concerning the hand. A new approach based on a measure of class separability is used in the feature selection. The paper shows that the Spare Steric Haar (SSH) features are effective for tracking the hands.

The process of real-time automate gesture recognition from an online video stream is the main objective of building a human-robot interaction system. The paper [12] succeeds in making an important step in achieving this goal by incorporating information on the estimated position and angles of the human body's joints.

An interesting approach to the gestures based control of vehicles is found in the work [13] which gave up a set of pre-defined gestures to control an UAV by pantomime gestures that mimic the actions of such a vehicle. This approach is more intuitive that would allow a human user to easily operate a robot.

This paper introduces a new interaction system between human and robot, called "Drive Me", and a new way to

978-1-5386-1495-2/18/$31.00 ©2018 IEEE

manipulate a electric robot, using dynamic gestures made with hands, not static positions. System performance analysis involved assessing the classification of the system, error rate, recall, sensitivity, and specificity.This paper is organized in 7 sections as follows: Introduction, Architecture of the "Drive Me" interaction system, Gesture set, Gesture recognition, Application functionality, Experimental results and discussions, and Conclusions.

## II. Architecture of the "Drive Me" Interaction System

The application is based on a Kinect sensor and a mobile robot Surveyor's SRV-1 connected to a computer with a Intel® Core(TM) 2 Quad Q9650 processor, running at a frequency of 3 GHz. The computer has 4 GB RAM memory.

### A. *The wireless robot Surveyor's SRV-1*

The aim of the interaction system between human and robot is to control by gestures a wireless robot, Surveyor's SRV-1, that is shown in figure 1. Designed for research, education and exploration, the robot Surveyor's SRV-1 is a mobile platform with a camera, a WiFi connection and a Blackfin processor. On the wireless robot runs a C compiled program, a mini operating system for the Blackfin processor. This firmware manages the exchange of information between the WiFi handler and can accept simple commands resulting the actions of the robot.

The operating system also contains a C interpreter, being possible to send C source code to be executed. The interpreted C language contains most control functions of the input- output streams, the repetitive structures from C, and also a set of functions to control the engines, the video camera, etc.

The SRV-1 robot uses the structure of the Blackfin camera with the BF537 processor with analog device at 500 MHz frequency, a video camera having a resolution from 160x128 to 1280x1024 pixels, a laser pointer or an optional ultrasonic field and a WLAN 802,11 b/g network attached to an engine with a mobile robotic base.

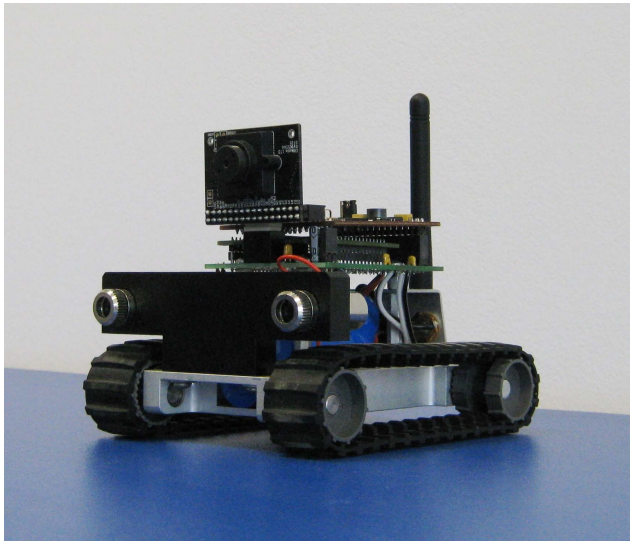

Fig. 1. The wireless robot Surveyor's SRV-1

### B. *The commands set*

All commands transmitted from the computer to the Surveyor's SRV-1 robot are strings of ASCII characters or ASCII decimal characters. All the commands are acknowledged from robot to host by a charcter "#" followed by the code of command. All the commands can be executed through a program incorporating TCP/Telnet communication capability. For instance, it can be connected using "netcat" by the command "nc robot-ip 10001". Travel commands for robot are not active until the engines have been initialized. For initialization the command "Mxxx" is used. A series of commands used for robot control are shown in Table I.

TABLE I. The commands set

| Command | Answer | Description |
|---|---|---|
| 8 | #8 | Forward |
| 9 | #9 | Right-forward |
| 2 | #2 | Back |
| 3 | #3 | Right-back |
| 4 | #4 | Left |
| 7 | #7 | Left-forward |
| 1 | #1 | Left-back |
| 6 | #6 | Right |
| 5 | #5 | Stop |
| 0 | #0 | The robot heads to the left by 20 degrees |
| . | #. | The robot heads to the right by 20 degrees |
| + | #+ | Increases engine speed |
| - | #- | Decreases engine speed |
| V | #Version…\r\n | Read the firmware version |
| $! | | Reset |

### C. *Software architecture*

Software architecture of the "Drive Me" interaction system consists of the following components:

- Microsoft Windows 7 32-bit operating system;
- Microsoft Visual Studio 2010 Professional;
- Microsoft Kinect SDK;
- Software application responsible with gestures acquisition, gestures recognition and sending orders to the robot.

Microsoft Kinect SDK includes Windows 7 compatible PC drivers for Kinect sensor. Kinect drivers for Windows 7 support:

- Kinect sensor microphone matrix as a kernel audio device that can be accessed through the standard Windows Audio API;
- Data stream for image and depth;

- Enumeration device functions which allow an application to use more than one Kinect sensor connected to the same computer.

Microsoft Kinect SDK provides a number of capabilities for software developers to write code for applications using programming languages such as C++, C# or Visual Basic, using Microsoft Visual Studio 2010. Microsoft Kinect SDK includes skeleton identification and tracking of one or two persons moving in the visible spectrum of the Kinect sensor. Also, the SDK allows access to data streams from the depth sensor, color sensor of the camera and from the four microphone matrices.

In the "Drive Me" system, we used Microsoft Kinect SDK to develop an application which achieves image acquisition and gesture recognition using the Dynamic Time Warping algorithm.

Once a gesture is recognized by the system as a valid command addressed to the robot, it is forwarded to it. For example, if the recognized gesture is "forward" it will be transmitted to the IP address of the robot the command with code 8, the command that makes the robot advance.

If the system recognizes the gesture being "behind" - then the command "2" it will be sent to the robot's IP address.The command "2" makes the robot to move back (behind). If the recognized gesture is "left" – then we will send to the IP address of the robot the command "4" – command that will cause the robot to move to the left.

If the gesture was recognized as "right" – then we will send to the IP address of the robot the command "6" – command that will cause the robot to move to the right. If the recognized gesture is "stop" then the command send to IP address of the robot is "5". This command causes the robot to stop.

## III. THE GESTURE SET

The "Drive Me" system uses a set of five gestures. The gestures are called: "go forward", "go back", "go to the left", "go to the right" and "stop".

The gesture set is shown in Table II.

TABLE II. THE GESTURE SET

| Gesture name | Gesture | Description |
|---|---|---|
| "Go forward" |  | The robot moves forward |
| "Go back" |  | The robot goes back |
| "Go to the left" |  | The robot moves to the left |
| "Go to the right" |  | The robot moves to the right |
| "Stop" |  | The robot stops |

Most gestures were performed by right hand, but we also have gestures performed with our left hand.

It was built an application that integrates the acquisition of gestures by the Kinect sensor, their recognition, and generation of the commands to the robot (corresponding to the recognized gesture), have been integrated into a single application. The workflow of the application is shown in figure 2. Gestures

were recognized using the DTW algorithm, and then the orders were sent to the robot.

## IV. GESTURE RECOGNITION

The gesture recognition system uses a classifier whose model was raised after a learning phase. At this stage of learning, a training set consisting of all the gestures used to control the robot was first made. The learning set includes gestures made by multiple users. Each description of a gesture, hereinafter referred to as pattern, has been analyzed and labeled with the class identifier to which the gesture belongs by a human expert. Patterns in the learning set represent different configurations of the human skeleton acquired at successive moments, and are the subject of pattern recognition techniques.

A human skeleton configuration is specified by the coordinates of six joints: the left shoulder, the left elbow, the left wrist, the right shoulder, the right elbow, the wrist. Every change in the skeleton position generates an event. The processing of this event is based on the records created in a file containing the coordinates values of the six joints.

In the recognition process, the gestured sequence acquired by the Kinect sensor are compared to the gesture sequence in the training set using the DTW (Dynamic Time Warping) algorithm. This algorithm calculates the minimum DTW distance between two coordinate sequences: the model sequence and the candidate sequence (to be a gesture). The DTW algorithm finds the best match between the two coordinate sequences (model and candidate sequence).

## V. APPLICATION FUNCTIONALITY

The workflow of the application is shown in figure 2.

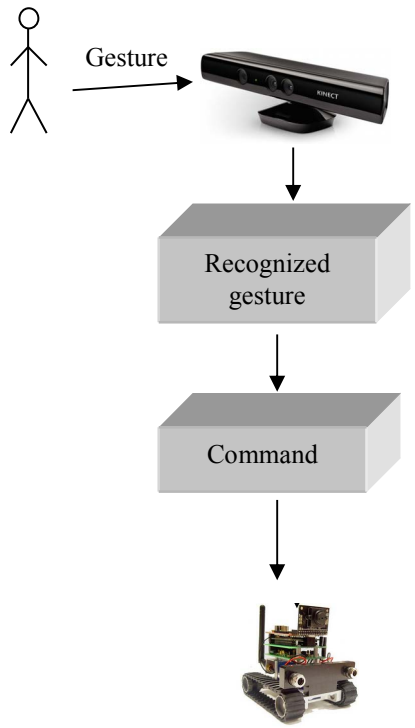


Fig. 2. The application workflow

As previously stated, the control of the robot is done by dynamic, motion gestures and not static postures, by processing the information flow from the Kinect device. Recognizing dynamic gestures in a continuous sequence of frames raises a number of issues. These are related to the fact that it is not known a priori either the beginning time of a gesture or its end. In addition, after a gesture has ended, the limbs of the human body return to a relaxed position. This move back to an initial position should not be recognized as a gesture. For these reasons, a window is slided over the continuous sequence of frames, and when a gesture is recognized in a sub-sequence, it is announced and then the corresponding command is sent to a robot (Algorithm 1, below).

In this detection of sub-sequences of interest in the frame sequence with information provided by Kinect, the dynamic programming method is used. The algorithm used is Dynamic Time Warping (DTW) that creates an initial matrix as the sequence S = {s1, s2 ,. . . } produced by Kinect, and calculates the distance from the gesture pattern set (gesture model) to be identified. Of course, a perfect fit between the model and the frames of the S sequence will not be found. For this reason, the algorithm will identify a gesture when an experimentally determined limit value is reached in the distance matrix.

The algorithm for connecting to the robot and transmitting commands to it, is presented in the following:

```
Algorithm 1:
1: robot address = 192.168.100.151
2: socket initialization
3: if the connection to the robot was successful then
4:      status = connected
5:      if recognized_gesture = "GO FORWARD" then
6:              send to the robot the command "8"
7:      end if
8:      if recognized_gesture = "GO BACK" then
9:              send to the robot the command "2"
10:     end if
11:     if recognized_gesture = "LEFT" then
12:             send to the robot the command "4"
13:     end if
14:     if recognized_gesture = "RIGHT" then
15:             send to the robot the command "6"
16:     end if
17:     if recognized_gesture = "STOP" then
18:             send to the robot the command "5"
19:     end if
20: else
21:     write "Error message."
22: end if
```

This algorithm receives the Id of the recognized gesture, provided by the DTW algorithm, elaborates and transmits the command to the robot.

## VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

The "Drive Me" system was tested using a total number of 500 gestures. For each class of gestures – "go forward", "go back", "go to the left", "go to the right" and "stop" - we stored in the training set, 100 patterns.

The evaluation of "Drive Me" system was made using the following performances parameters: classification accuracy, precision, recall rate, sensitivity and specificity.

A classified pattern is considered as an example that can be either positive or negative. Each of the decision of the classifier may be in of the following distinct categories: TP (true positive), TN (true negative), FP (false positive) and FN (false negative).

True positive (TP) are those patterns that belongs to a class $C_x$ and are classified in class $C_x$. True negative (TN) corresponds to the negative examples correctly identified as negative. False positive (FP) refers to those negative examples that are incorrectly identified as positive. False negative (FN) refers to positive examples incorrectly identified as negative [14].

After performing the tests, we obtained, for each gesture, the values from Table III.

TABLE III. THE CONFUSION MATRIX

| | **Go forward** | **Go back** | **Go to the left** | **Go to the right** | **Stop** |
|---|---|---|---|---|---|
| TP | 63 | 64 | 66 | 53 | 69 |
| TN | 371 | 400 | 396 | 400 | 400 |
| FP | 29 | 0 | 4 | 0 | 0 |
| FN | 37 | 36 | 34 | 47 | 31 |

Based on Table III, it is possible to evaluate some performance parameters such as: accuracy of gesture classification, precision, recall, sensitivity and specificity.

**Accuracy of gesture classification**

A common strategy to evaluate a gestural interaction system is to calculate the accuracy of gesture recognition and the error rate.

The accuracy of the gesture classification refers to the proportion of correctly identified gestures relative to the total number of gestures. The error rate is the proportion of incorrectly classified gestures, relative to the total number of gestures. The better the classifier it is, the higher its accuracy and the lower its error rate.

As it is known, the accuracy rate of gesture classification was calculated using the formula (1) and the error rate was calculated using the formula (2):

$$Accuracy = (TP+TN)/(TP+TN+FN+FP) \quad (1)$$

$$Error\ rate = (FP+FN)/(TP+TN+FN+FP) \quad (2)$$

In Table IV are presented the values of the accuracy for each gesture.

TABLE IV. ACCURACY AND ERROR RATE FOR GESTURE

| **Gesture** | **Accuracy** | **Error rate** |
|---|---|---|
| Go forward | 86.80 % | 13.20 % |
| Go back | 92.80 % | 7.20 % |
| Go to the left | 92.40 % | 7.60 % |
| Go to the right | 90.60 % | 9.40 % |
| Stop | 93.80 % | 6.20 % |

**Precision of gesture classification and recall rate**

The precision of gesture classification is the ratio of the number of correctly classified gestures reported to the number of all gestures classified in that class.

The recall rate is the ratio of the number of correctly graded gestures relative to the all the gestures number from that class.

Precision is a measure of the quality of gesture classification, taking into account the correctness of their classification. The recall rate measures the utility of the classification by quantifying the proportion of the relevant results.

Generally, a high rate of recall indicates that very few irrelevant results will be provided by the classifier.

The precision of gesture classification was calculated using the formula (3) and for recall it was used the formula (4):

$$Precision = TP/(TP+FP) \quad (3)$$

$$Recall = TP/(TP+FN) \quad (4)$$

Precision and recall obtained for every gesture class are presented in Table V.

TABLE V. THE VALUES OF PRECISION AND RECALL FOR GESTURE

| **Gesture** | **Precision** | **Recall** |
|---|---|---|
| Go forward | 68.5 % | 63 % |
| Go back | 100 % | 64 % |
| Go to the left | 94.3 % | 66 % |
| Go to the right | 100 % | 53 % |
| Stop | 100 % | 69 % |

**Sensitivity and specificity of gesture classification**

Sensitivity is the ratio between the number of positive samples recognized by the system as positive and the total number of positive samples. For example, the percentage of "Go forward" gesture that are correctly recognized as "Go

forward". Sensitivity refers to the ability of the recognition system to identify true positive samples.

If a system has a sensitivity of 1.00, this means that the system correctly recognizes all positive samples. For example the system recognizes all "Stop" gestures as "Stop" gestures. A system that has a high sensitivity also has a low error rate. Sensitivity is calculated using the formula (5) and specificity with the formula (6):

$$Sensitivity = TP/(TP+FN) \quad (5)$$

Specificity measures the proportion of negative samples that are correctly identified by the system as negative (for example, the percentage of gestures that are not the "Go forward" gesture and are correctly recognized as not "Go forward".

An optimal theoretical prediction aims to achieve a sensitivity of 100% and a specificity of 100%. Specificity refers to the system's ability to identify true negative samples. A system that has a high specificity rate will have a low error rate. Specificity is also known as the true negative rate and it is calculated using the following formula:

$$Specificity = TN/(TN+FP) \quad (6)$$

The values corresponding to the sensitivity and specificity obtained for each gesture are shown in Table VI.

TABLE VI. THE VALUES OF SENSITIVITY AND SPECIFICITY FOR EACH GESTURE

| Gesture | Sensitivity | Specificity |
|---|---|---|
| Go forward | 63 % | 93 % |
| Go back | 64 % | 100 % |
| Go to the left | 66 % | 99 % |
| Go to the right | 53 % | 100 % |
| Stop | 69% | 100% |

## VII. CONCLUSIONS

In this paper, we introduced a new gesture interaction system, called "Drive Me", a system that controls the movements of a wireless robot through hand gestures, in real time.

Also, we have made an analysis of the performance of the "Drive Me" system. We have calculated some performance parameters: accuracy of gesture classification, error rate, system's precision, rate of recall, sensitivity and specificity.

The main contributions of the paper are: a new proposed interaction techniques (using dynamic hand gestures), integration of the two applications (the gesture recognition application and the application that sends commands to the IP address of the wireless robot) and performance analysis of "Drive Me" system.

The experimental results show that "Drive Me" is a robust system, with a high gesture recognition rate. The systems works in real time and it is independent of the lighting conditions, the clothes of the user that makes the gestures and environment. The distance from the user who makes the gestures to the robot, or to the Kinect sensor, does not make difficult the gesture recognition.

The "Drive Me" system works both in daylight and at night, thanks to the Kinect sensor which has an infrared sensor.

Several users tested the system and they had a positive user experience.

## REFERENCES


[1] http://humanrobotinteraction.org/1-introduction/

[2] O. M. Vultur, Ş. G. Pentiuc and A. Ciupu, "Navigation system in a virtual environment by gestures," *2012 9th International Conference on Communications (COMM)*, Bucharest, 2012, pp. 111-114. doi: 10.1109/ICComm.2012.6262541

[3] O. M. Vultur, S. G. Pentiuc and V. Lupu, "Real-time gestural interface for navigation in virtual environment," *2016 International Conference on Development and Application Systems (DAS)*, Suceava, 2016, pp. 303-307.doi: 10.1109/DAAS.2016.7492592

[4] O. M. Vultur and S. G. Pentiuc, "Navigation System in Virtual Environments Using Human Gestures," National Conference Distribuited Systems, Suceava, 2011, pp. 11-14.

[5] R. Agrawal and N. Gupta, "Real Time Hand Gesture Recognition for Human Computer Interaction," *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, Bhimavaram, 2016, pp. 470-475. doi: 10.1109/IACC.2016.93

[6] E. G. Craciun, L. Grisoni, S.G.Pentiuc, I. Rusu, "Novel Interface for Simulation of Assembly Operations in Virtual Environments", Advances in Electrical and Computer Engineering. 13. pp. 47-52. 10.4316/AECE.2013.01008.

[7] T. Cerlinca, S. G. Pentiuc and V. Vlad. "Real-Time 3D Hand Gestures Recognition for Manipulation of Industrial Robots." *Elektronika ir Elektrotechnika* [Online], vol.19 no.2, 2013.

[8] S.G. Pentiuc, O.M. Vultur, and A. Ciupu "Control of a Mobile Robot by Human Gestures. " In: Zavoral F., Jung J., Badica C. (eds) Intelligent Distributed Computing VII. Studies in Computational Intelligence, vol 511. Springer, Cham, 2014.

[9] Jinguo Liu, Yifan Luo, and Zhaojie Ju, "An Interactive Astronaut-Robot System with Gesture Control", Comput Intell Neurosci. 2016; PMID: 27190503, 2016: 7845102. doi: 10.1155/2016/7845102

[10] Radu-Daniel Vatavu, "Beyond Features for Recognition: Human-Readable Measures to Understand Users' Whole-Body Gesture Performance", International Journal of Human–Computer Interaction, 33:9, 713-730, DOI: 10.1080/10447318.2017.1278897

[11] C. Liu, Y. Y. Chen, and L. C. Fu, "Robust dynamic hand gesture recognition system with sparse steric haar-like feature for human robot interaction," 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Tsukuba, 2016, pp. 148-153.doi: 10.1109/SICE.2016.7749213

[12] A. Guler, N. Kardaris, S. Chandra, V. Pitsikalis, C. Werner, et al.. "Human, Joint Angle Estimation and Gesture Recognition for Assistive Robotic Vision". Gang Hua ; HervéJégou ACVR, ECCV, Oct 2016, Amsterdam, Netherlands. Springer, 9914, pp.415 - 431, 2016, LNCS

[13] M Burke and J Lasenby, "Pantomimic gestures for human–robot interaction", IEEE Transactions on Robotics 31 (5), 1225-1237

[14] P. Falinouss "Stock Trend Prediction Using News Articles: A Text Minning Aproach", http://epubl.ltu.se/1653-0187/2007/071/LTU-PB-EX-07071-SE.pdf

[15] A. M. Faudzi, M. H. Kuzmani, M. A. Azman, and Z. H.Ismail,"Real-time Hand Gestures System for Mobile Robots Control", Procedia Engineering, Vol 41, 2012, 798-804, ISSN 1877-7058

[16] Junyun Tay and Manuela Veloso, "Modeling and Composing Gestures for Human-Robot Interaction", Proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication, Versailles, France, pp. 107-112, Sept. 2012