The 18th IEEE International Symposium on
Robot and Human Interactive Communication
Toyama, Japan, Sept. 27-Oct. 2, 2009

Gestural teleoperation of a mobile robot based on visual recognition of sign language static handshapes

C. Tzafestas[1], N. Mitsou[1], N. Georgakarakos[1], O. Diamanti[1], P. Maragos[1], S.-E. Fotinea[2] and E. Efthimiou[2]

[1] School of Electrical and Computer Engineering, National Technical University of Athens
Zographou Campus, 15773, Athens, Greece
ktzaf@softlab.ntua.gr, maragos@cs.ntua.gr

[2] Institute for Language and Speech Processing, ILSP - ATHENA R.C.
Artemidos 6 & Epidavrou , Maroussi GR-15125, Greece
evita@ilsp.gr, eleni_e@ilsp.gr

Abstract— **This paper presents results achieved in the frames of a national research project (titled "DIANOEMA"), where visual analysis and sign recognition techniques have been explored on Greek Sign Language (GSL) data. Besides GSL modelling, the aim was to develop a pilot application for teleoperating a mobile robot using natural hand signs. A small vocabulary of hand signs has been designed to enable desktop-based teleoperation at a high-level of supervisory telerobotic control. Real-time visual recognition of the hand images is performed by training a multi-layer perceptron (MLP) neural network. Various shape descriptors of the segmented hand posture images have been explored as inputs to the MLP network. These include Fourier shape descriptors on the contour of the segmented hand sign images, moments, compactness, eccentricity, and histogram of the curvature. We have examined which of these shape descriptors are best suited for real-time recognition of hand signs, in relation to the number and choice of hand postures, in order to achieve maximum recognition performance. The hand-sign recognizer has been integrated in a graphical user interface, and has been implemented with success on a pilot application for real-time desktop-based gestural teleoperation of a mobile robot vehicle.**

I. Introduction

In the frames of a national research project (Acronym: "DIANOEMA", Full Title: "Visual Analysis and Sign Recognition for Sign Language Modelling and its Application in Robot Teleoperation"), and beyond, we are considering applications of multi-modal human-machine interfaces, incorporating vision-based human interaction modalities by means of natural and intuitive (hand, body or facial) gestures. In this context, a pilot application has been developed that concerns hand-gestural teleoperation of a mobile robotic vehicle. The first step was to design an appropriate "vocabulary", which consists of a small set of hand signs (for the time being, static hand postures) that constitute a robot command language. A "desktop" teleoperation scenario was selected, as illustrated in Fig. 1, where the gestural commands of the human operator are issued remotely, from a master control station that supports all the necessary computer vision gesture recognition operations.

For the purposes of our first gestural teleoperation pilot application, this "desktop-type" typical arrangement of remote human-robot control was preferred to a "robot-centered" scenario, where human-operator and robot platform interact in a more direct way, in a co-located space and time. Scenarios of the latter type have recently been reported in the literature, with on-board visual systems of robot platforms tracking "whole-body postures" and interpreting them as desired actions (commands) of the human operator. Such work has for instance been carried out in the Virtual Reality and Active Interfaces Laboratory of EPFL, where a prototype stereo visual recognition system has been developed for a small set of basic motion commands inferred in the form of static body postures [20]. Related work has also been reported in [6], where the development of an interface recognizing seven basic static gestures, appropriately chosen for teleoperation. Similar systems have also been presented in [5], [19], [2].

The work presented in this paper focuses on recognizing hand-postures as a natural interaction modality for conveying human intention to robot commands. This is in line with the objectives of the DIANOEMA research project focusing on developing computer-vision methods and algorithm towards analyzing, modelling, and recognizing natural sign language utterances. A desktop-type teleoperation has been considered more appropriate for this type of human-robot interaction. A multi-level teleoperation architecture has been considered, inspired by related work in the field of telerobotics [15], [10], [18]. The system supports: (a) low-level, direct teleoperation commands, (b) mid-level shared-autonomy commands (based on autonomous, sensor-based robot behaviors, and (c) high-level operations (e.g: <go-to-room><#value>), which are implemented and included in the command set. For the first pilot application scenario, a high-level teleoperation sequence was implemented, which consists of issuing a command of the above third type (autonomous mode of operation). Autonomous mobile-robot navigation algorithms have been implemented and tested experimentally, including: (a) path-planning, (b) collision avoidance, and (c) continuous localization and motion correction (based on static geometric landmarks). Experiments have been conducted at the premises of the Robotics Laboratory of our Department,

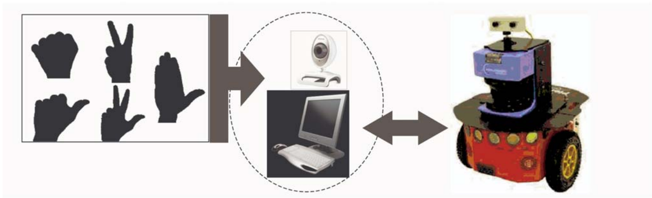


Fig. 1. *Gestural, sign teleoperation application scenario*

using a Pioneer P3-DX indoor mobile robot platform.

The paper is organized as follows. Section II describes the specification of the sign robot commands vocabulary, and Section III presents the image analysis methodology followed to extract pertinent handshape features. The MLP neural-network (NN) classifier, implemented to recognize hand postures as intended teleoperation commands, is described in Section IV, where experimental results are also presented. The graphical user interface integrating all these modules in the considered gestural teleoperation scenario is presented in Section V, and the paper concludes with Section VI, where future work directions are also highlighted.

II. Design of Hand-Sign Teleoperation Vocabulary

Our first objective was to analyze the specifications with respect to the considered robot teleoperation pilot application, and to define a set of robot commands that will constitute the gestural telerobot control vocabulary. Considerations regarding architectures of telerobotic control and robot teleoperation control modes were taken into account and motivated the design. Natural sign language data were also analyzed, and a set of signs (static hand postures) was finally selected to constitute this small vocabulary for gestural robot teleoperation control.

A. Teleoperation Commands - Telerobotic Control Modes

Analyzing the control modes that can be supported by a telerobotic system, in relation to properties involving a more or less direct human-robot interaction, or ease of control and the degree of tele-cooperation between the human operator and the telerobot, we have defined a language for mobile robot teleoperation control with the robot motion commands structured in three levels of control. This structure fits into a general multi-level telerobotic architecture, in accordance with concepts related to supervisory and shared-autonomy telerobot control.

Level-1: "Low-Level" - Direct Teleoperation.

Level-1 refers to commands addressing low-level robot control, directly affecting the driving mechanisms of the mobile robot vehicle. These commands are:

1-0: <initialize sequence> (i.e.: GET READY!)
1-1: <move-forward> #meters [OR <until condition>]
1-2: <rotate> (<right> [OR <left>]) #value (e.g. angle)
1-3: <start-motion> (GO!)
1-4: <pause-motion> (PAUSE!)
1-5: <abort-motion> (STOP!)

These commands thus give the human operator the ability to obtain direct control of the motion performed by the teleoperated mobile robot. Although this control level offers the advantage of a direct interaction between human and robot, it has major disadvantages (particularly in the presence of large time delays in the bilateral communication and control loop) related to the mental workload required by the human operator, which may lead to a deterioration of the system's performance. Furthermore, this control level is not suitable for exploiting at a full extent the capacities of the robot to perform some (local or global) navigation functions in an autonomous (sensor-driven) way, and may potentially even lead to posing safety issues (e.g. with respect to collisions with static or dynamic obstacles).

Level-2: "Mid-Level" motion commands - Teleoperation based on sensor-driven autonomous robot behaviors.

Robot commands at this level refer to a set of autonomous robot control functions (sensor-driven behaviors). These behaviors require some form of "local intelligence" embedded on the robotic vehicle, reasoning on data captured from on-board sensors. The human-operator is then in charge of commands based on and exploiting the presence of these autonomous robot behaviors, while keeping high-level decision making responsibilities. For instance, the human operator conducts the global path planning operations, while the on-board robot controller performs tasks like local path planning and collision avoidance and tracking of environmental geometric features (like walls, in indoor environments, etc.) for localization purposes.

The commands that are implemented in the current prototype version of the system are basically the following:

2-1: <follow-wall> <left> [OR <right>]
2.2: <pass-through-doorway> <left> [OR <right> OR <front>]

Level-3: "High-Level" commands, fully-autonomous robot navigation.

This level incorporates commands that require increased degree of robot autonomy and intelligence embedded on the mobile robot vehicle, based on perception and reasoning functions like global path planning and route finding, besides local obstacle avoidance. These commands relieve the human operator from tasks associated to motion control of the robot platform. The role of the operator is now to "indicate global intentions" (like the target-position of the mobile robot's motion), and to perform some form of high level supervisory control (by means of visual, real-streaming and/or simulated/graphical, feedback, and high-level commands intervention).

The robot commands considered at this level of control are of the following basic types:

3-1: <go-to-room> #value (e.g. room number)
3-2: <follow-me> (in the case of a "robot-centered" application scenario).

All the above control operations form a list of available

TABLE I

LIST OF ROBOT COMMANDS - ROBOT TELEOPERATION VOCABULARY

Basic Commands	GET_READY; GO; PAUSE; STOP (ABORT); #numerical values
Low-Level	MOVE_FORWARD; ROTATE; LEFT; RIGHT;
Mid-Level	FOLLOW_WALL; UNTIL_CONDITION; PASS_THROUGH; DOORWAY; HALLWAY;
High-Level	GO_TO_ROOM; FOLLOW; FOLLOW_ME;

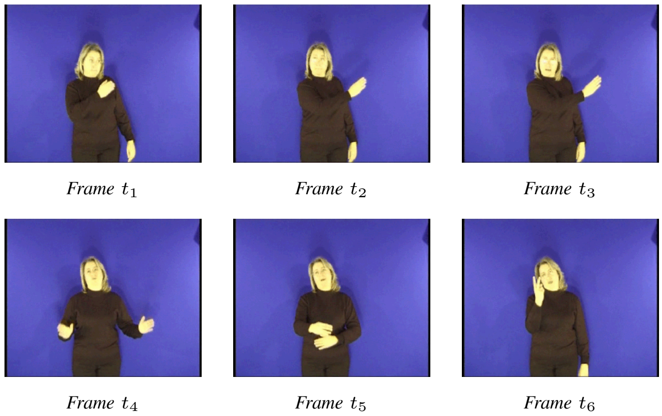


Fig. 2. *An example of a complete sign robot-command utterance: Command of the high-level type <GO-TO> <ROOM> [#number].*

robot commands, constituting the robot vocabulary that is to be considered for association with gestural (hand-signs) commands. This "Vocabulary of Robot Commands" is summarized in Table I.

B. *Hand Signs as Telerobot Commands*

In the frames of the DIANOEMA project, a video-corpus of an indicative subset of the Greek Sign Language (GSL) was created and annotated, comprising: (a) a list of lemmata that are representative of the use of hand-shapes as a primary sign formation component (developed on the basis of measurements of hand-shape frequency of use in sign morpheme formation); (b) a set of controlled utterances, which form paradigms capable to expose the mechanisms GSL uses to express specific core grammar phenomena; and (c) free narration sequences, which are intended to provide data of spontaneous language production that may support theoretical linguistic analysis of the language and can also be used for machine learning purposes regarding sign recognition. In this context, a set of video sequences was recorded and analyzed, concerning a number of lemmata that refer to pre-defined robot teleoperation commands. Our goal here was to study the morphology of such hand sign commands issued by natural signers, and derive useful guidelines regarding the hand-shapes that can be selected for the defined small vocabulary of telerobot commands. Fig. 2 illustrates a sequence of instances showing utterance of a complete robot command of the high-level type (Frames t_1-t_3: <GO-TO>, Frames t_4-t_5: <ROOM>, Frame t_6: [#NUMBER 2]).

Let us point out here that the set of suggested signs used by different natural signers participating in the project, for the predefined robot teleoperation command utterances, presented very little variations, meaning that in general, different signers were congruent as to the handshapes to be employed for the specified robotic vocabulary. The video corpus that was created was then analyzed to identify the type of signs employed regarding robot command utterances. Our goals in the design of this sign vocabulary were twofold: (a) to encode the robot commands using a set of static signs (handshapes) that can be reliably recognized in real-time by available visual processing and image analysis methods, and (b) to approach as much as possible the characteristics of the natural sign utterances, as recorded and analyzed in the video corpus.

Based on the above requirements and analysis results, we have designed a small set of robot teleoperation commands. For the first pilot application scenario presented in this paper, high level supervisory telerobot control is performed, with commands of the type <GO-TO> [#region-ID-number]; that is, commands instructing the robot to move into one of a set of predefined target-locations. Autonomous robot navigation functions, like path planning, collision avoidance, and localization have been employed, but are beyond the scope of this paper and are not further analyzed here.

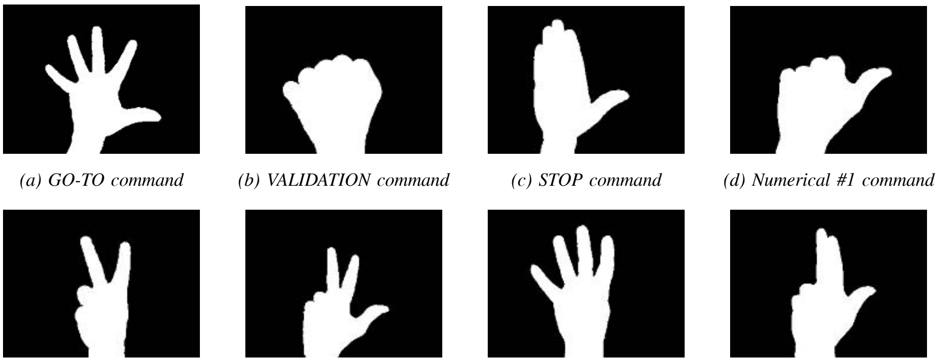
Fig. 3 shows the set of hand postures selected for the first pilot application. In the considered application scenario, the human operator (in a desktop-like teleoperation environment) issues a sequence of commands, that can take the following form:

1) <GET_READY> command, that is, initialization of the telerobotic system to receive motion commands;
2) <MOVE_TO> command;
3) #Numerical command, indicating the number (id) of the predefined target-location where the robot is instructed to transit.

In each step of this procedure, after each hand-posture sign command is issued by the human operator and properly recognized by the system, the system (by means of a graphical user interface) must respond indicating the current state of the system together with the sign command that is being recognized. When the utterance of a teleoperation command, consisting of the above three steps, is completed, the system awaits for user confirmation, before the actual robot motion command is finally sent to the mobile robot platform for execution. User "validation" thus constitutes the final (fourth) step in this gestural teleoperation procedure:

4) <VALIDATION> command: confirmation of recognized motion command (GO!).

Robot motion can then be interrupted by the human operator at any time, by issuing a <STOP> command (handshape (c) in Fig. 3). In the prototype system developed for the first pilot application described in this paper, the same handshape is used both for the <GET_READY> and <VALIDATION>



(a) GO-TO command *(b) VALIDATION command* *(c) STOP command* *(d) Numerical #1 command*

(e) Numerical #2 command *(f) Numerical #3 command* *(g) Numerical #4 command* *(h) (unassigned command)*

Fig. 3. *The list of static handshapes used as robot commands in the first pilot application.*

command (handshape (b) -fist- in Fig. 3). A finite state machine implements this procedure, where various parameters can be tuned, like the number of instances needed for the system to consider a handshape properly recognized (for safety and robustness purposes).

Conclusively, for the first pilot study we have opted for a supervisory, "teleprogramming" type of high-level telerobotic control, assuming sufficient intelligence (planning and navigation capabilities) embedded on-board the mobile robot platform. This forms an adequate application scenario for this proof-of-concept pilot experiment, where our initial objectives were to illustrate the basic principles of employing sign language commands in remote mobile robot control. The small vocabulary of robot commands will be enriched in the near future to incorporate (a) a larger set of commands following the design principles outlines above, and (b) dynamic gestures, where time characteristics of sign utterances are also going to be taken into account in the recognition process.

III. Visual Handshape Features Extraction

A. Hand Posture Image Segmentation

The first stage for the processing of a sign video stream concerns the segmentation of each image frame, for the detection and tracking of the hand postures. For segmentation purposes, we first apply a skin color segmentation procedure, based on a prior statistical model of human skin color. Further processing employing geodesic active contour models gives excellent segmentation results. However, to comply with the real-time constraints, such PDE-based models are not used in the pilot application presented in this paper.

The skin color segmentation procedure finally integrated in the pilot system is similar to the one presented in [1]. More specifically, using the probabilistic skin color image derived based on the prior skin color model, an iterative double-thresholding process is applied to identify skin color pixels, satisfying either one of the two following conditions:

(a) The skin color probability in a specific pixel is greater than a threshold T_1.

(b) The skin color probability in a specific pixel is greater than a threshold $T_2 < T_1$, AND there exists at least one neighboring pixel that is assigned as skin color.

Indicative values for the two thresholds are: $T_1 = 0.5$ and $T_2 = 0.1$. Results of applying this technique are shown in the second column of Fig. 4.

At a next stage, the resulting skin segmentation images undergo a set of filtering processes, to eliminate any existing erroneous pixels and to produce segmented areas that have a smooth contour, so that reliable extraction of shape features can be subsequently enabled. More specifically, the images are firstly processed by an area opening filter to reject any small segmented regions that may have been erroneously identified as skin. Then, a series of morphological opening / closing filters is applied to smooth out the segmented regions further. An area opening is then applied to keep the largest connected component in the image as the final segmented binary hand image. The contour curves of the resulting segmented hand image may be further processed through Gaussian filtering to produce a more natural segmentation result. Indicative results are illustrated in the third column of Fig. 4.

The statistical color model used is created from a large set of skin images that have been manually extracted a-priori. Nevertheless, for robustness purposes, this model can be dynamically adapted by updating the statistical skin color model on-line, exploiting the new information available after each frame segmentation. More specifically, the segmented hand-posture area in each image frame is used to update the color model values. The procedure involves color transformation into the Lab space, isolation of a, b coefficients, and incrementing of the respective entries in the color model table. By applying this dynamic updating, the results are improved, particularly enhancing the robustness of the procedure for different signers in varying lighting conditions.

B. Shape Descriptors Computation

Various shape descriptors can be considered as inputs in a handshape recognizer system. These include Fourier descriptors (e.g. [21], or [7] for a gesture recognition application), histograms [14], n*th*-order moments (e.g. like performed in [16]), curvature scale space (e.g. [12] or [4] for gesture recognition application), etc. In the work reported in this paper, we have explored two different types of handshape

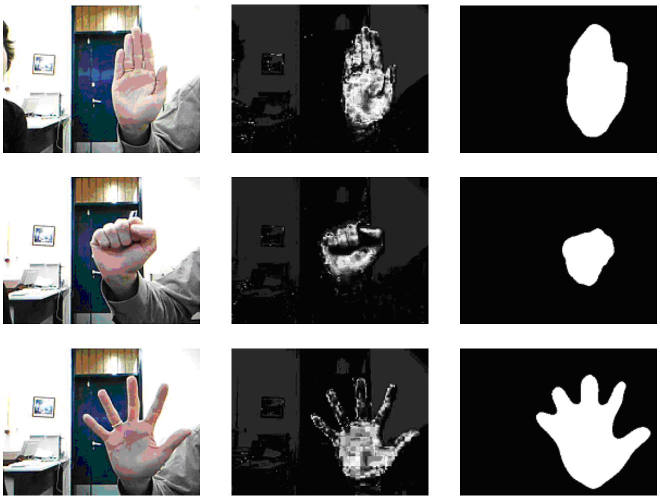


Fig. 4. *Indicative results of skin color modelling and handshape segmentation. First column shows original image, second column depicts skin color probabilities, and third columns final segmented binary images.*

descriptors: (a) *boundary-based* descriptors, extracting features related to the form of the curve defining the contour of the segmented handshape, and (b) *region-based* descriptors, related to properties of the enclosed segmented region.

1) Region-based features: The region-based features we have explored are based on the computation of the basic moments of the segmented handshape area. These features are used quite often in shape recognition problems, because they achieve very big information compression rates (since the use of only a small number of these moments may be adequate to describe a shape). Based on the computation of such moments, various combinations of shape descriptors can be defined that are inherently invariant to affine transformations.

In this work, we have implemented and experimented with the following descriptors:

- the zero*th*-order moment, that is, the total handshape area a;
- the two first-order moments, yielding the coordinates of the center of mass (x, y);
- the eccentricity e, which can be formally defined as: $e = \left[(\mu_{20} - \mu_{02})^2 + 4\mu_{11}\right]/a$, and gives a measure of how "elongated" is the shape, where μ_{ij} $(i, j = 1, 2)$ define the centered second-order moments;
- the degree of compactness c, defined approximately as the ratio of the area of the segmented handshape to the area of a circle having the same perimeter: $c = (4\pi a)/(\text{perimeter})^2$ (giving a measure of how "circular" is the shape).

2) Boundary-based features: Computation of such features is performed on the contour of the segmented image. We used two classes of features:

- The first class was based on the computation of the curvature of the contour; a typical representation uses the histogram of the curvature.
- The second class was based on the computation of Fourier coefficients on a coordinates of the contour.

Fourier descriptors proved experimentally to be particularly important, for handshape recognition in our application setting. The procedure implemented to compute these shape descriptors is as follows: First, the segmented handshape contour is extracted (after applying an automatic "wrist-cropping" procedure). A maximum of $N = 256$ points are taken on the contour (depending on how long is the perimeter of the shape). DFT descriptors can then be defined on these N points using the formula:

$$Z_k = \sum_{n=0}^{N-1} z_n e^{-2\pi jkn/N}$$

The first coefficient Z_0 is ignored, to make the shape descriptors translation invariant. The magnitudes of the rest of the coefficients are taken, normalized by Z_1, making the descriptors invariant to rotation and scaling.

IV. Sign Robot Command Recognition with a Multi-Layer Perceptron Classifier

The set of features described in the previous section is used as input to a neural-network (NN) classifier, aimed to recognize the segmented image from the set of available handshapes, constituting the robot sign command vocabulary described in Section II.

Several hand-gesture recognition methods are reported in the literature. Chang et al. [3] used a nearest-neighbor technique, together with the definition of a distance measure for a template matching based classification. Isaacs and Foo [8] used a two-layer NN to recognize signs from the American Sign Language (ASL). Juang et al. [9] used a fuzzy recursive Takagi-Sugeno-Kang NN (FTRFN) for a gesture recognition problem. In [17], orientation histograms on grey-level images define the features used as inputs to an MLP NN gesture recognizer. In [13], the use of an RBF NN is proposed in combination with a Hidden Markov Model (HMM) for a gesture recognition application. A robot teleoperation application is reported in [11], where a simple heuristic was applied limited to the recognition of 5 handshapes, interpreted as basic robot motion commands (of the form: forward, backwards, right, left, and stop).

The NN classifier that has been implemented in the work reported in this paper consists of an MLP with one or more hidden layers and 8 nodes in the output layer (each node in the output layer corresponds to a handshape). A small set of 60 images were used as the training set (7 to 8 examples for each one of the 8 robot command handshapes, captured from 5 different signers). Learning rate for the back-propagation algorithm was experimentally chosen equal to 0.3, with 2000 epochs as maximum learning period. The hyperbolic tangent sigmoid activation function was used, while the Nguyen-Widrow layer initialization function was employed regarding weights and biases.

Various conditions were tested experimentally, including different sets of input feature vectors, as well as different number of hidden layers and nodes. A set of 20 successive experiments were performed with random initial weights

TABLE II

HANDSHAPE NN RECOGNITION RESULTS FOR DIFFERENT INPUT CONFIGURATIONS

13 Fourier descriptors	additional 13 Fourier	e	c	wrist cropping	Recognition Rate
$\surd$	−	−	−	auto	83%
$\surd$	$\surd$	−	−	auto	88%
$\surd$	$\surd$	$\surd$	$\surd$	auto	91%
$\surd$	$\surd$	$\surd$	$\surd$	manual	98.5%

TABLE III

COMPUTATION TIMES FOR DIFFERENT SHAPE FEATURES

Image Features	Computation Time
Fourier	40 msec
Eccentricity + compactness	35msec
Curvature Histogram	510 msec
Auto wrist cropping	130 msec

for each one of the conditions tested. Test data consisted of 24 images (3 examples for each handshape). The shape features were normalized so that the inputs supplying the NN classifier take values ranging from -1 to $+1$, with standard deviation 1 and mean value equal to zero. The experimental analysis conducted led to the following conclusions:

- The use of only one hidden layer is adequate in our application context, for classifying the given 8 handshapes. Differences in performance achieved with the addition of extra hidden layers are not statistically significant.
- The optimal number of nodes in the hidden layer is approximately double the number of the inputs nodes.
- For real-time applications, as the gestural robot teleoperation context considered in this paper, the Fourier descriptors seem to be of particular importance, providing a good trade-off between computation speed and recognition rate.

Table II summarizes the experimental results A sufficiently good result, in terms of recognition rate and real-time performance for our application context, has been obtained with only 13 inputs (Fourier shape descriptors), 30 neurons in the hidden layer and 8 nodes in the output layer (83% recognition rate). Applying a two-scale morphological filtering in the images and extracting 13 additional Fourier descriptors (giving us a total of $13+13 = 26$ inputs for the NN classifier) leads to an increase in the performance (recognition rate of 88%). Adding another two inputs (being the eccentricity e and compactness c features, defined above) leads to an additional slight performance improvement (recognition rate of 91%).

It must be pointed out here that despite the small number of images (7 to 8 images per hand posture) used each time at the training set, the system performance was quite satisfactory. Initial experimental analysis has showed us that, given the complete data set used in our study, any number above 5 for the size of the training set, was adequate to train the system satisfactorily. Of course, our goal in the future is to populate the data set used for training the NN classifier (also using captures from the GSL video corpus developed), and then capture live data in different conditions, in order to validate quantitatively the performance of the system under real (hard) operating conditions. The effect of noise in the images has also been explored, both in the case of natural (due to segmentation errors) and artificially introduced noise. In both cases, the system performance was maintained, showing that the method with the considered 26 inputs (2×13 Fourier descriptors) is quite robust.

It must be also noted at this point that the above results are obtained with the application of an automatic wrist cropping procedure, used after handshape segmentation and morphological filtering to extract and keep only the hand area, disregarding any segmented area from the forearm. It is observed, though, that automatic wrist cropping procedures may fail in specific handshape configurations. This may affect recognition results, since part of the handshape may also be cropped in some cases. If we perform manual wrist cropping on all segmented images, which is equivalent to imposing a "long sleeves assumption" (i.e. assuming that the human signer uses non-skin colored clothes with long sleeves), often considered as a requirement in automatic gesture recognition settings, then the results obtained are considerably improved (recognition rate of 98.5%), as shown in Table II. The computation time for feature extraction, using an Intel Celeron 1.46GHz processor with 512MB RAM, are shown in Table III. The computation of 26 Fourier descriptors with eccentricity and compactness features leads to a near real-time system, and forms our choice of preference for the considered robot teleoperation pilot application.

V. PILOT IMPLEMENTATION IN MOBILE ROBOT SIGN TELEOPERATION

The handshape segmentation and recognition modules described above have been integrated within a graphical user interface (GUI), designed for our mobile robot teleoperation pilot application. The GUI incorporates all communication functions for: (a) submitting teleoperation commands to the remote mobile robot platform (in our case, a Pioneer P3-DX robot vehicle), and (b) receiving information feedback from the onboard robot sensors (camera, ultrasound, odometry). Fig. 5 shows an instance of this GUI, where the handshape recognition modules are integrated, including a hand-signer live video window, a handshape segmentation window, and a command recognizer panel (indicating the sign robot command currently recognized by the system). A simple finite-state machine implements the robot command sequence, as this has been specified in Section II. The system has been successfully tested in real-time experiments, in the considered "desktop" gestural teleoperation scenario.

VI. CONCLUSION AND FUTURE WORK

This paper described a gesture-based desktop teleoperation system and its pilot application for the teleoperation of a mobile robot. The paper focused on image analysis problems and on visual handshape recognition, which is

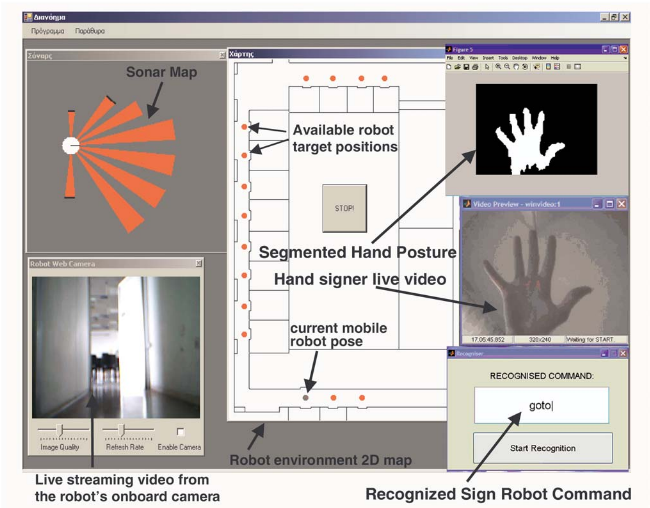


Fig. 5. *Graphical User Interface incorporating the handshape recognizer for the mobile robot teleoperation application scenario.*

here performed using a neural-network classifier. A small vocabulary of telerobot commands has been defined, and has been associated to a set of static hand-postures, thus forming the robot teleoperation sign language subset for our pilot application. A series of experiments has been conducted and results are presented that identify important image-based handshape features that can satisfy both real-time constraints and acceptable recognition rates. A desktop teleoperation scenario has been implemented, integrating all the telerobot communication and control modules within a graphical user interface.

Robot teleoperation, and more generally human-robot interaction, based on natural gestures like sign language utterances, constitutes a very interesting and challenging field in robotics, with important currently on-going research activities worldwide. Our future research efforts in this domain will focus on several directions:

- To enrich the vocabulary of robot commands, incorporating a larger set of signs, that are potentially closer to natural sign language utterances.
- To implement and integrate dynamic hand gesture recognition algorithms, incorporating motion tracking and Hidden Markov Models.
- To explore different human-robot interaction scenarios, involving multimodal communication that also uses human gestures (and generally actions) inferred from motion of both hands and arms.

VII. ACKNOWLEDGMENTS

This work was partially funded by the European Union, in the framework of the Greek national project DIANOEMA (GSRT, M3.3, id 35).

REFERENCES

[1] A. A. Argyros, M. I. A. Lourakis, Tracking Skin-colored Objects in Real-time, invited contribution to the *Cutting Edge Robotics* Book, ISBN 3-86611-038-3, Advanced Robotic Systems International, 2005.
[2] V. Bonato, A. K. Sanches, M. Fernandes, J. M. P. Cardoso, E. Simoes, and E. Marques, "A Real Time Gesture Recognition System for Mobile Robots", In: *International Conference on Informatics in Control, Automation, and Robotics* (ICINCO'04), August 25-28, Setubal, Portugal, 2004, INSTICC, pp. 207-214.
[3] C.-C. Chang, I.-Y. Chen, Y.-S. Huang, "Hand pose recognition using curvature scale space", In 16th International Conference on Pattern Recognition, Volume 2, 11-15 Aug. 2002, pp:386 - 389, vol. 2.
[4] C.-C. Chang and C.-M. Pengwu, Gesture recognition approach for sign language using curvature scale space and hidden markov model, *in 2004 IEEE International Conference on Multimedia and Expo (ICME'2004)*, June 2004 pp:1187 - 1190, Vol.2
[5] H.-M. Corradini, "Camera-Based Gesture Recognition for Robot Control", In: *IEEE-INNS-ENNS International Joint Conference on Neural Networks* (IJCNN'00), Volume 4, p. 4133, 2000.
[6] A. Hu, M. Q. Meng, P. X. Li, and X. Wang, "Visual Gesture Recognition for Human-Machine Interface of Robot Teleoperation", in: *Proceedings of the 2003 IEEE Intl. Conf. on Intelligent Robots and Systems* (IROS'2003), pp. 1560-1565, 2002.
[7] C.-L. Huang and W.-Y. Huang, Sign language recognition using model-based tracking and a 3d hopfield neural network, *Mach. Vision Appl.*, 10(5-6):292-307, 1998.
[8] J. Isaacs, and S. Foo, "Optimized wavelet hand pose estimation for American sign language recognition", *In 2004 Congress on Evolutionary Computation (CEC'2004),* Volume 1, 19-23 June 2004, pp:797-802, Vol. 1.
[9] C.-F. Juang, K.-C. Ku, S.-K. Chen, "Temporal hand gesture recognition by fuzzified TSK-type recurrent fuzzy network", *In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks (IJCNN '05)*, 2005.
[10] A. Kheddar, C. Tzafestas, P. Coiffet, T. Kotoku, and K. Tanie, Multi-Robot Teleoperation Using Direct Human Hand Actions, *International Journal of Advanced Robotics*, Vol. 11, No. 8, pp. 799-825, 1997.
[11] A. Malima, E. Ozgur, and M. Cetin, "A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control," *In the 14th IEEE Signal Processing and Communications Applications Conference*, 17-19 April 2006.
[12] F. Mokhtarian and A.K. Mackworth, A theory of multiscale, curvature-based shape representation for planar curves, *IEEE Transactions on PAMI*, 14(8):789-805, August 1992.
[13] C. W. Ng, S. Ranganath, Real-time gesture recognition system and application, *Image and Vision Computing*, 20, 2002, p. 993-1007
[14] E.J. Ong and R. Bowden, "A boosted classifier tree for hand shape detection", *in AFGR'04*, pp. 889-894, 2004.
[15] T. B. Sheridan, *Telerobotics, Automation and Human Supervisory Control*, The MIT Press, 1992.
[16] T.E. Starner, J. Weaver, and A.P. Pentland, Real-time american sign language recognition using desk and wearable computer based video, *IEEE Transactions on PAMI*, 20(12):1371-1375, December 1998.
[17] K. Symeonidis, *Hand Gesture Recognition Using Neural Networks*, M.Sc. Thesis, School of Electronic and Electrical Engineering, Surrey University, August 2000.
[18] C. S. Tzafestas, Virtual and Mixed Reality in Telerobotics: A Survey, In: *Industrial Robotics: Programming, Simulation and Applications*, Advanced Robotic Systems International, ISBN 3-86611-286-6, pp.437-470, Feb. 2007.
[19] J. Wachs, U. Kartoun, H. Stern, and Y. Edan, "Real-time hand gesture telerobotic system using fuzzy c-means clustering", n: *Proceedings of the 5th Biannual World Automation Congress*, 2002, volume: 13, page(s): 403- 409.
[20] C. Wengert, B. Poell, S. Grange, T. Fong, and R. Siegwart, *Human Oriented Tracking and Mobile Robot Gesture Driving*, Technical Report, EPFL (Swiss Federal Institute of Technology), Virtual Reality and Active Interfaces Group, 2002.
[21] C.T. Zahn, R.Z. Roskies, Fourier Descriptors for Plane Closed Curves, *IEEE Transactions on Computers*, Vol. c-21, No. 3., March 1972, pp. 269-281.