International Journal of the Physical Sciences Vol. 6(18), pp. 4478-4491, 9 September, 2011
Available online at http://www.academicjournals.org/IJPS
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Speech control of a teleoperated mobile humanoid robot

Baocheng Wang[1], Zhijun Li[1]* and Weijun Ye[2]

[1]Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China.
[2]School of Navel Architecture Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China.

Accepted 15 July, 2011

This paper presents a mobile humanoid robot platform which is able to understand humans' speech commands in teleoperation environments. For service in unstructured environments, the robot must operate efficiently under active auditory perception system to ensure a coordinated human-robot system. First, the speech-based teleoperation control is introduced, with the cameras mounting on the robots, transmitting the video to the user through the wireless network, while the user sends speech commands to drive the robot to fulfill the desired task through the same communication channel. In order to eliminate the time delay in the communication channel, the authors incorporate the event-based motion control into the robot control. It drives the system to achieve the best possible motion. Finally, the event-based speech teleoperation is experimentally implemented and verified using a human-like mobile robot.

Key words: Human-like mobile robot, speech recognition, teleoperation, event-based control.

INTRODUCTION

In recent years, intelligent machines play more and more important role in human life. For example, in Kim et al. (2010), the 3D motion analysis system helps clinicians and engineers in the prevention, diagnosis and treatment of people with physical disorders. Besides, humanoid robots' serving potential evokes much interest. Particularly, intelligence service robots are the main type of the humanoid robot that has been studied (Takahashi et al., 2010; Cielniak et al., 2010; Mitsunaga et al., 2006). To serve a human being, it is necessary to develop an active auditory perception system for the robot system that can execute various tasks in everyday environments under the natural human speech. Moreover, to adapt to more complicated and varied environments, the robot is equipped with more and more sensors, and in order to unify the coordinates of them, calibration is indispensable, e.g. the typical hand-eye calibration (Li et al., 2009). More importantly, the robot needs to be operated at a high speed with stability. The active auditory perception is very essential for robots to be able to interact with humans and their environments. When the human and the robot work together, there are two extremes for initiative in the interaction. On one hand, the robot can be seen as a tool for accomplishing specific tasks by human commands. The obvious human-robot interface is teleoperation: the human gives commands and the robot carries out the requested actions. On the other hand, a robot may work largely autonomously, with a human acting as a supervisor, intervening only when things go wrong. In autonomous robots, the capability of navigation is essential to sense the surroundings, perceive the working environment, plan a trajectory and execute proper reaction using the information (Nakhaeinia et al., 2011; Albaker and Rahim, 2011) and in Banga et al. (2011), the authors have shown optimal control for a robot arm using fuzzy logic and genetic algorithms. For many robotic applications, as the tasks to be performed by robots grow more complex, an interactive interface incorporating communication technique, such as natural language dialogue, can provide a more intuitive and flexible means of coordinating human-robot activity. Several recent systems have been developed that permit

*Corresponding author. Email: zjli@ieee.org.

natural-language human-robot interaction. These include both task-based systems like the NASA peer-to-peer human robot system (Fong et al., 2006), as well as more social, embodied systems such as Leonardo (Breazeal et al., 2004) and JAST dialogue robot (Foster et al., 2009). In many of these systems, the domain or conversational role of the human and the robot system are predefined and distinct, and while there is possibility for joint or collaborative activity, it is constrained to specific tasks, the user may help the robot's vision system to identify the correct target, or the user may instruct the robot to perform a task. Due to the possibilities offered by the network technology and the new tools that offer new possibilities to manage the remote environment, a voice-based teleoperation scheme that allows interaction with the remote environment in a more comfortable and flexible way is presented in this paper. Working over a classical teleoperation environment, the goal is to reach a higher level of abstraction in the user commands. The tool allows the operator to interact with the remote environment through natural language recognition. This system is able to interpret and execute the commands formulated by the operator in natural language, according to the elements present in the remote environment. Natural language programming offers a great possibility to the operator to easily control and manage the remote environment (John et al., 2010; Veres and Lincoln, 2008). Voice processing might establish a natural dialogue between a human and the robot in the remote environment in such a way that a not-qualified user of the system can manage the robot through a semantics that represents the environment and its relationship with the robot. The built robot described in this paper allows the operator to interact with the remote environment through natural language recognition. We know natural language interfaces are not adequate for all kinds of robotic commands. There exist languages or human-friendly interfaces which are not natural but which are more suitable for particular applications. Specifically, in the applications presented in this paper, it is shown that natural language input can be an efficient technique for high level commands and other input interfaces are more appropriate for low level commands.

Speech recognition and natural language processing becomes a powerful tool for human-computer communication. In Fan and Li (2010), a commercial speech recognition chip was used to control robot prosthesis. In Schuller et al. (2009), the speech control was used in laparoscopic surgery to operate the manipulator. However, these works all focus on the specific applications or in specific fields. In Thorission (1999), a communicative humanoid robot with a hand and graphic face was presented. The robot appears on a small monitor in front of the user. It is capable to perform face-to-face dialog, in real time, with a human user with various hand gestures, facial expressions, body language and meaningful utterances, which can be used to guide the humanoid-like robot. In Matthew et al. (2009) and Alessandri et al. (2005), natural language and gesture understanding was integrated to give a more natural interface in space and medical robotic applications. In Lu et al. (2010), a speech recognition system was incorporated to a teleoperated humanoid robot which was applied to daily tasks. In Reinoso et al. (2007), the voice assistance tool is presented to interact with the robot in a natural way and integrate real-time feedback in natural language. In Reinoso et al. (2007), a spoken dialogue interface to a mobile robot working in an office environment, which a human can direct to specific locations, ask for information about its status, and supply information about its environment. Speech-based control was all implemented for the local operation rather than the remote operation (Thorission, 1999; Matthew et al., 2009; Alessandri et al., 2005; Bos and Oka, 2007) and in the teleoperation implementation (Lu et al., 2010; Reinoso et al., 2007) assume that the command transmission is without time delay and the networked teleoperation is under the perfect network condition with ignoring the time delay. However, it is well-known that the network delay and packet dropouts can degrade the system performance or even cause instability of control systems. Therefore, how to handle the network induced delays and packet dropouts for the speech-based teleoperation has attracted much attention, moreover, to our best knowledge, under the realistic commands transmission environment, the proposed voice control approach in these works cannot be applied to the remote teleoperation. Therefore, in this paper, we shall introduce the speech-based teleoperation control of a mobile humanoid robot under a general network conditions. As for the problem of the time delay which the network brings with, we incorporate and modify the event-based scheme (Xi et al., 1996) to handle the problem. To validate our system, we design many tasks for the robot to complete. The robot can move around in environments and perform physical tasks, such as searching objects and sending live video and audio information, providing kinds of information for the users. The robot is partly autonomous, and it carries out its missions in the immediate and shared environments. The user, on the other hand, is busy with his/her ordinary activities while communicating with the robot. In this paper, we enhance the efficiency of the service robot by developing speech control scheme. First, the speech-based teleoperation control is introduced. The cameras mounted on the robots transmit the video to the user through the network, and while the user sends speech commands to drive the robot fulfill the desired task through the same communication channel. In order to eliminate the time delay in the communication channel, we incorporate the event-based motion reference for the robot. It drives the system to achieve the best possible motion. Finally, the event-based speech teleoperation is experimentally implemented and tested for the control of human-like

Figure 1. A mobile humanoid robot.

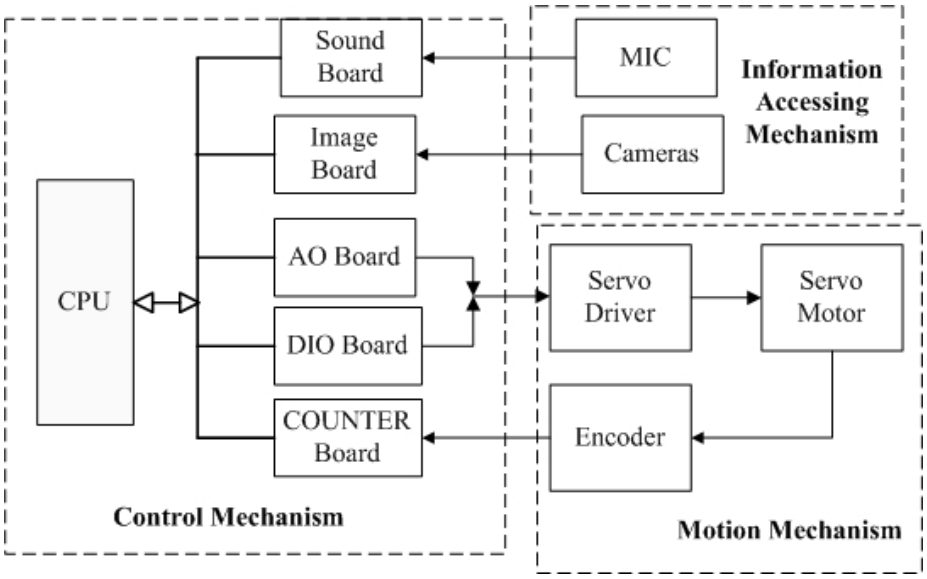

Figure 2. The Structure of mobile humanoid robot.

mobile robot with good performance.

SYSTEM DESIGN AND OVERVIEW

A humanoid robot platform

The built-up mobile humanoid robot is shown in Figure 1. The structure of the mobile humanoid robot can be divided into three modules: the control module, the motion module and the vision-auditory perception module, as shown in Figure 2. The control module is comprised of a control computer with many necessary boards, such as an image processing board, a sound board, a DA board, a DIO board and a counter board. The motion module consists of 8 servo motors and the

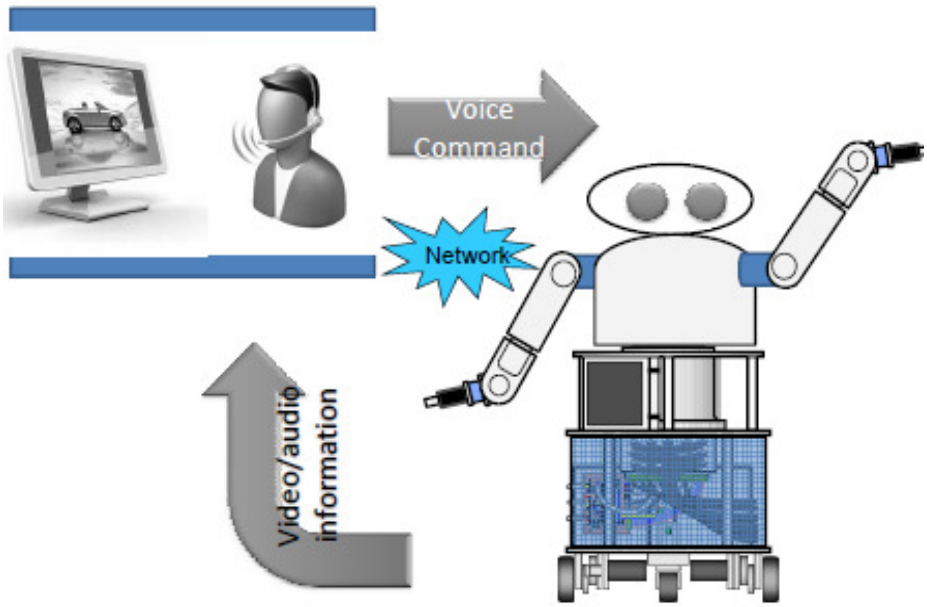

Figure 3. The speech-based teleoperation system.

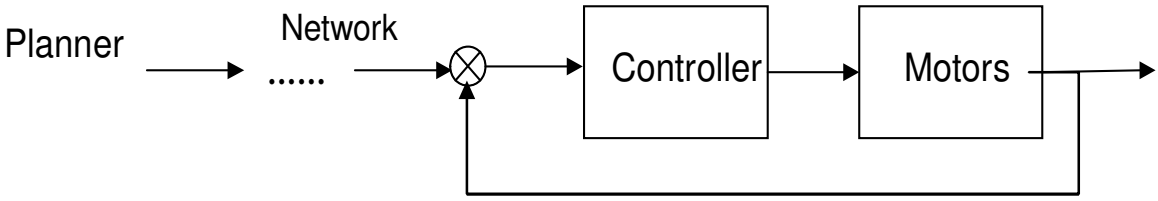

Figure 4. Traditional remote control.

corresponding servo drivers as well as the encoders. For the vision-auditory perception module, two monitoring cameras and a microphone are mounted on the shoulder. This robot is driven by two independent wheels and consists of two arms, and each one is composed of 3 joints.

Event-based teleoperation

The speech-based teleoperation system is shown in Figure 3. In teleoperation, the operator and the robot are connected via a communication channel, which often involves large distance and imposes limitation for data transfer between the local and the remote sites. Therefore, the time delay may happen between the time when a command is generated by the operator and the time when the command is executed by the remote robot. It is well known that time delay in the communication channel can destabilize the whole teleoperation system if it is not well compensated (Kang et al., 2010). It is even worse for the networked teleoperation (e.g. internet-based teleoperation), since in this case, the delays are stochastic, irregular and unsymmetrical (Li et al., 2011). Therefore, we propose the event-based speech control to handle the time delay. Traditional networked teleoperation systems can schematically be described as shown in Figure 4. The core of the system is the feedback control loop, which ensures the system stability and robustness. The feedback brings the control with the time delay information, which brings the instability into the system. For the speech-based teleoperation, the time delay would include the speech recognition and network transmission. In the proposed structure shown in Figure 5, a new motion reference variable is different from time, but directly related to the speech measurement of the system. Instead of time, the speech commands are parameterized by the new motion reference variable. The motion reference variable is designed to efficiently carry the speech information needed for the planner to modify the original plan to form a desired output. As a result, for any given time instant, the action plan is a function of the

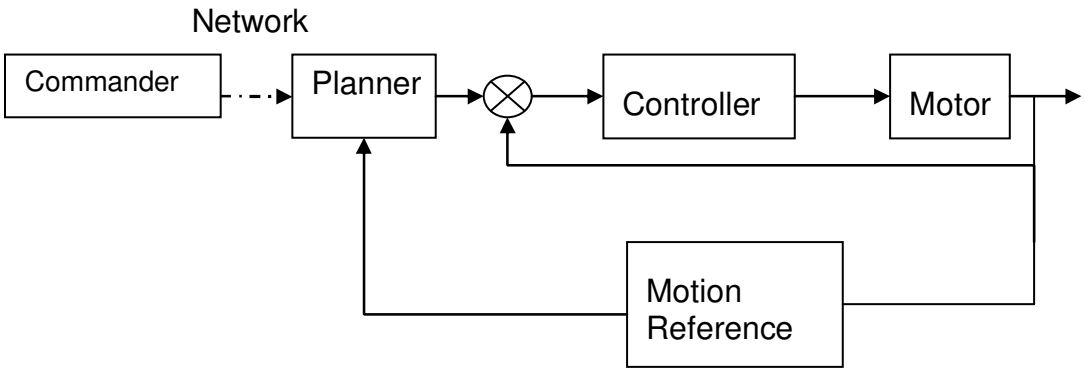

Figure 5. Remote event-based control.

system output. This creates a mechanism to adjust and modify the plan based on the speech measurement. More importantly, it makes the planning a closed-loop real-time process. The details of the implementation are discussed subsequently. The local motion controller for the robot is proportional integral derivative (PID) control. For simplicity, we only use the proportional control. The control procedure of the system is described as the DA board sends the analog signals to the motor drivers according to the commands received from the controller. While the digital input/output (DIO) board can only receive and output digital signals, it will be used to control the rotation direction and to stop the motor when necessary. The counter board, working as the feedback mechanism, can receive the number of pulses of the encoder while the motor is working. Through this way, we can calculate the current position of the motor. The image board and the sound board are for the video and audio information.

Remote video/audio transmission

Two kinds of socket technology can be used to create a video/audio data transmission, the stream socket and the datagram socket. The former can provide a connection-oriented, sequenced and unduplicated flow of data with well-defined mechanisms for creating and destroying connections and detecting errors. It is confirmation based, meaning it transmits data and waits for confirmation from the other side. It not, it retransmits. In this paper, we use the datagram socket, which provides a connectionless data transmission, and it does not guarantee the data sequenced and unduplicated. However, it is easy to build and the packets transmitted are independent and much smaller than the ones in the former. Since it does not need to guarantee the correctness of the data, the efficiency of the datagram socket can be better than the stream one. The comparison between the two techniques is fully described (Munir, 2001) and in the authors' experiments, the delay in the stream socket ranges from the minimum value of 100 ms to as much as 3000 ms, while in the datagram socket, the delay ranges from minimum value of 100 ms to the maximum value of 250 ms, with rare data loss. So we can tell that the datagram socket is much better for teleoperation. In our case, we design the video socket, the text socket and the audio socket, respectively, making each one simple. The work flow is shown in Figure 6. Since the video transmission data is huge and important, the video compression standard, H.263, is used. First it subtracts the former frame from the current frame, and only the residual will be encoded. Moreover, the motion estimation module is used to reduce the information further, and through the DCT (discrete cosine transformation) transform, we can transform the remaining frame data to coefficients in time domain, and then discard the value which is close to zero and encode them by an entropy encoder. Similarly, the decoding process is the revision of the encoder one. As we all know, in the encoder, we discard the value which is close to zero, so it cannot be restored in the decoder. The scale of the raw image we gather from the image card is 704 × 576, and we use this image for our image processing which is necessary in object searching. For real-time transmission, we reduce the image to CIF format (that is, 352 × 288), and we can obtain 15 frames per second. The whole working procedure video transmission is simply shown in Figure 7. Here, we introduce the speech recognition for a teleoperated mobile humanoid robot. We use the Microsoft speech SDK, which is free, to develop our human-robot speech interface. The Microsoft speech SDK contains many functions for developers to create a robust and easy-to-use speech interface including recognition and speech synthesis. The speech application interface (SAI) provides a high-level interface between physics systems and speech engines, which implements all the low-level details needed to control and manage the real-time operations of various speech engines. Firstly, the recognition engine needs to be initialized, and then the main interface of recognition can receive various kinds of recognition notification. Finally, a

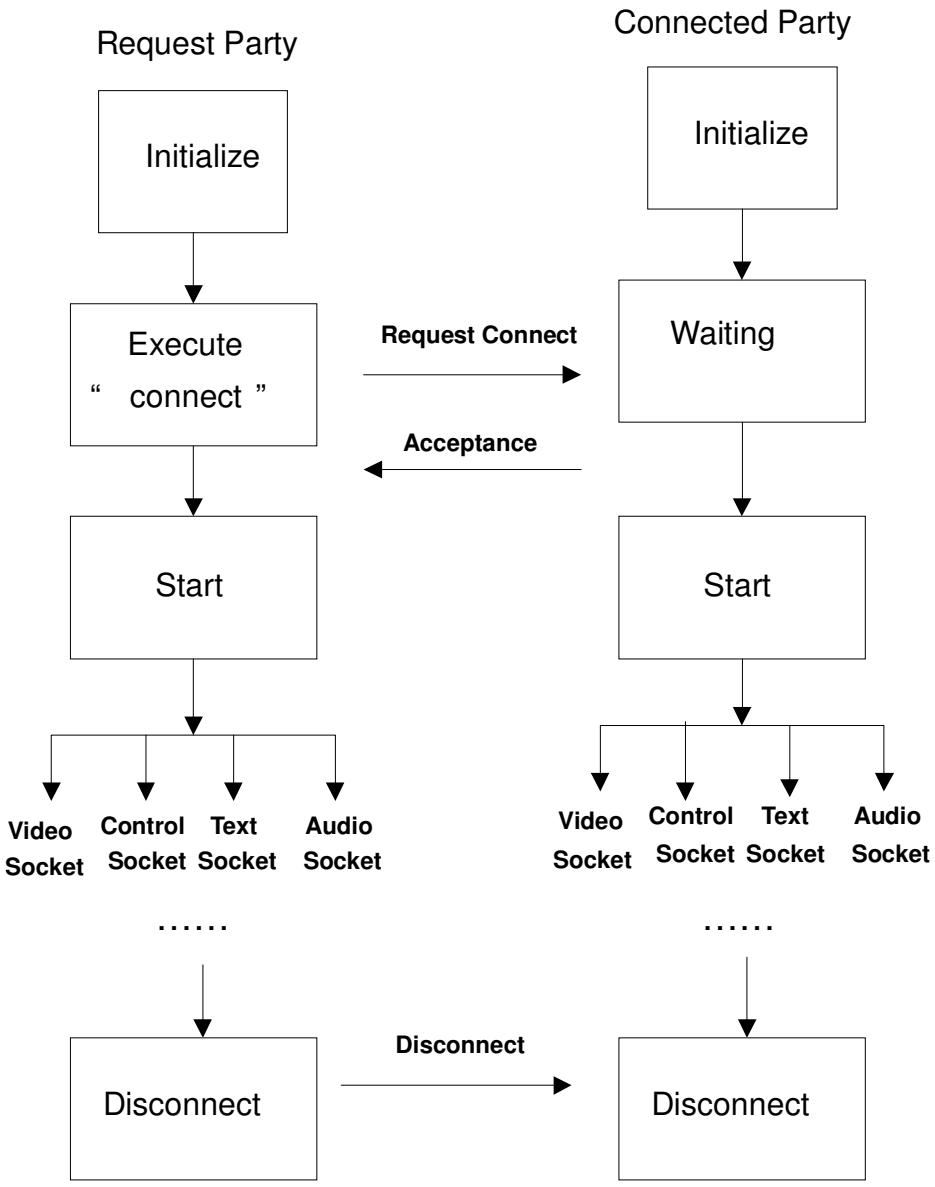

Figure 6. The network structure.

speech application must create, load and activate an instance, which essentially indicates what type of utterances to recognize, that is, a dictation or a command and control grammar. The dictation grammar contains rich words and phrases, but it has very low recognition accuracy. In this paper, we use the command and control grammar, and the grammar can be edited in the XML format file. Then we design an optional train step, which contains three parts as the MIC train, the environment train and the word addition. When the program runs, it waits for recognition. Once accepting the recognition notification, we can decide what command the user has given and switch to the corresponding routine. Since human's language is so rich that it is very hard to predict what the user would say, we choose the commands with the highest frequency. For example, in order to make the robot run, the user may say “run” or “begin”, and even attaches some other words which may not make some significance but can make the user more comfortable, such as “at once”, or “please”. Considering these problems, we design the command file as follows:

```
<RULE NAME="RUN", TOPLEVEL="ACTIVE">
<L>
 <P>run</P>
 <p>begin</p>
</L>
 <o>at once</o>
 </RULE>
```

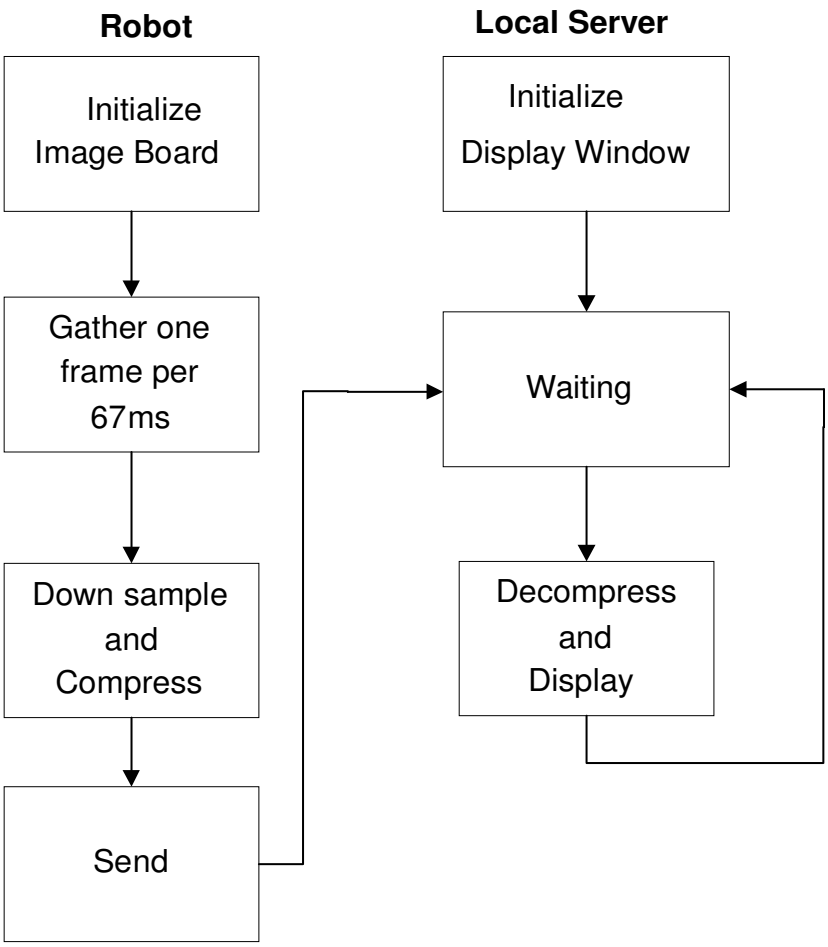

Figure 7. Video transmission.

Moreover, we set these works “run”, “begin”, or “run at once”, “begin at once” with the same response. For simplicity and accuracy, we classify the “forward” command. If we only say “forward”, the command meaning is “forward sixty centimeters”, and if we say “go straight”, the command meaning is “forward ten meters”. Once these words are recognized, we can switch to the corresponding event response. If we want the robot to distinguish the word within a phrase to trigger a rule, we propose the following scheme. For example, if we want to distinguish clearly between “turn right” and “turn left”, we can add a “+” before the word “right”:

<P>turn +right</P>

OBJECT SEARCHING

Object searching can be divided into two different parts. First, we have to determine the position of the object and then through analyzing the features of that region, we can track it. The first part can be completed using SURF (speeded up robust features) (Bay et al., 2008), and the second is with camshift algorithm (Bradski, 1998). The SURF algorithm consists of two parts: the interest points’ detector and the descriptor. The approach for detecting interest points is a Hessian-matrix approximation using integral images, which reduces the computation time drastically. As for the descriptor, in order to be invariant to rotation, it identifies the main and reproducible orientation. For the purpose, it calculates the Haar wavelet responses in x and y direction within a circular neighborhood of the interest point, and then a sliding orientation window of size $\frac{\pi}{3}$ is used. Through summing the abscissa and ordinate responses in the window, a local orientation vector is yielded, and it chooses the longest as the major orientation. For the extraction of the descriptor, a square region centered on the interest point is constructed and oriented along the selected orientation. The region is split into 4 × 4 square sub-regions. Within each sub-region, Haar wavelets responses in 2×2 sub-divisions and their sums are

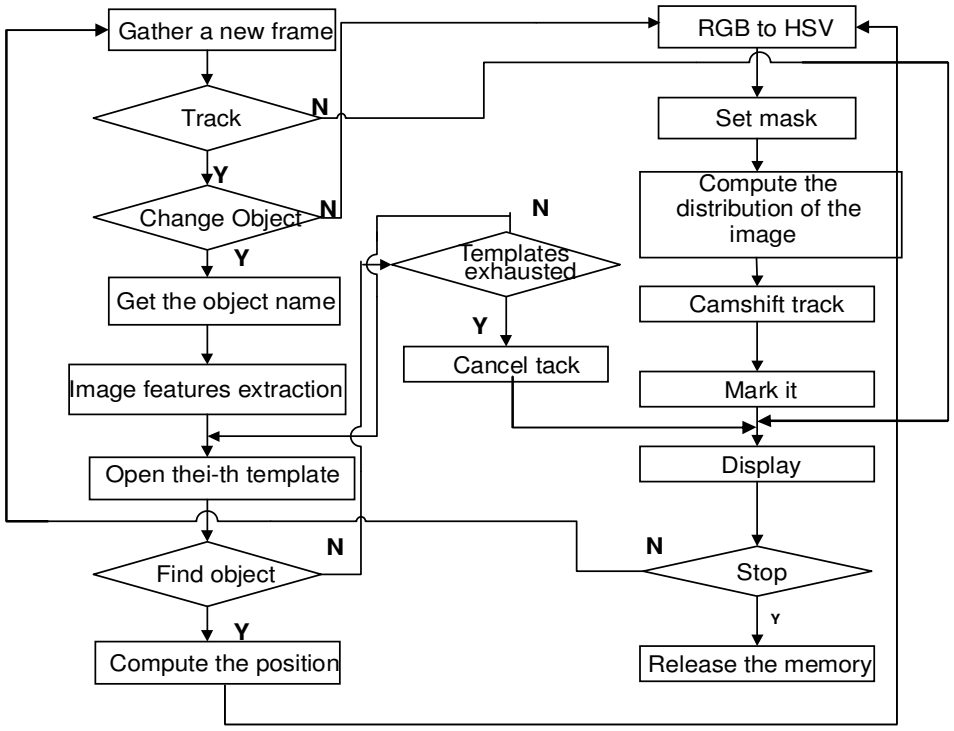

Figure 8. Object searching work flow.

computed, that is, $\sum d_x$, $\sum |d_x|$, $\sum d_y$, $\sum |d_y|$. Finally, we have a vector of length 4 × 4 × 4 (that is, 64). In the match step, we use approximately nearest neighborhood search to find the matched points. The proposed method to identify the object is to gather some pictures of an object in kinds of viewpoints. When we get a new frame, we search in the corresponding set to determine whether there is object or not. Camshift is based on another algorithm called meanshift, which is a stable way to find the local extremum in density distribution of a set of data. Simply, at first, the zero order moment M_{00} and one order moments M_{10} , M_{01} are calculated according with Equation 1.

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \quad (1)$$

Then the center of the new region can be figured out as follows:

$$(x_c, y_c) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right) \quad (2)$$

The procedure of mean-shift is in the following:

1. Select the initial search window.
2. Compute the weighted center of the window.
3. Set the center of the window in the computed center.
4. Return to step 2 until the position of the window unchanged. Camshift is a little different from mean-shift, because the scale of the search window can adjust automatically. The length and width of the tracking window are calculated out as follows (Bradski, 1998)

$$L = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (3)$$

$$W = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (4)$$

where $a = \frac{M_{20}}{M_{00}} - x_c^2$, $b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right)$ and $c = \frac{M_{02}}{M_{00}} - y_c^2$.

Above all, the two methods are based on the color and texture of the tracking region, so it asks the image is color. While SURF relies on the gray scale, by using both of them, we can make full use of the image information. The whole work flow is illustrated in Figure 8. Firstly, the object position in the image is determined by the SURF part through matching templates, and after finding the object, the camshift part will tack its position in the image automatically.

Table 1. Amount of the compressed data.

13036	*12890*	*12478*	*12401*	*12156*	*12768*

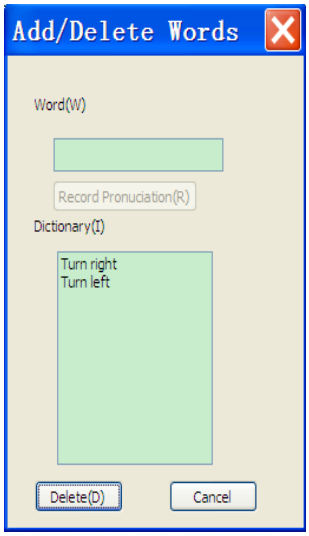

Figure 9. Word addition UI.

EXPERIMENTS

Network transmission

In order to validate the performance of the robot, some extensive experiments are conducted, and the programming platform is based on OpenCV 2.1 library. First, for defining the video buffer size, we testify the compress ratio of H.263. We use the high resolution image (704 × 576) for the image processing, and down sample it to low resolution (352 × 288) for compressing and transmitting. The amount of raw data is constant: C=352 × 288 × 3=304128 bytes. The value of some compressed data is listed in Table. 1. Therefore, we can allocate the appropriate memory space. In the experiment, we allocate 15000 bytes for compressed data and 350000 bytes for the raw data. The compress ratio can be obtained as follows:

$$R = \frac{6C}{13036+12890+12478+12401+12156+12768} = 24.09698 \quad (5)$$

Thus, we can see that after compressing, the need of bandwidth for the network is dramatically decreased from 3 m to 0.12 m. In this condition, the high efficiency of the datagram socket can be fully reflected.

Control interface

Here, we will give a detailed description about our event-based control scheme. As mentioned, (Lu et al., 2010; Reinoso et al., 2007), they all ignore the delay which the network induced. In Reinoso et al. (2007) the authors even introduce the real-time feedback through the network, which may work in the local or free network, but it is probable to fail in the busy network, and specially, in our network, the video transmission is a huge burden for the network, as a result, the network is congested all the time. In our control scheme, the commands give out his order and trigger corresponding routines. As in Figure 5, we can see that the planner is implemented on the robot part instead of the remote control part. As a result, it is immune to the network delay and it is based on a motion reference variable related with the output of the system instead of the time, that is, the output of the planner is changed according to the output of the system, so this structure is called event-based control. Then we define speech commands, and it is easy to train, which includes three parts: the MIC training, the environment training and the word addition, for example, if you find some word has very low recognition rate, which may means that it is not in the default dictionary, and then you can use the word addition, as shown in Figure 9. The command is read, and the recognizer will record it. The display results of the user interface are shown in Figures 10 and 11. The defined commands are listed in Table 2. The Dialog is demonstrated in Table 3. Then, we will verify the speech command function. After the “begin” is recognized and sent, the robot driven program will be run by “ShellExecute” function, and through “FindWindow” function, we get the window handle, then we can send messages to the “Robot” window.As for the robot turning, going forward and going backward control is based on the turning angles of the two wheels. So we will have a first discussion about the angle coordinates of the robot, which is like Figure 12. When the robot is started, the angle origin is set, and it will not change during the runtime. Therefore, we have to send the absolute angles to the robot. But it will be inconvenient to record the angle every time. We try to make this work automatic. We set eight variables to store the current angles of the two wheels (that is, left_angle_l, left_angle_r, right_angle_l, right_angle_r, up_l, up_r, down_l, down_r). When any one of them is changed, the others should also change. If the robot turns right at 45°, it is equivalent that the right wheel goes backward 84.6 mm, and the left wheel goes forward the same distance. The evolution of angles is listed in Table. 4. Based on the predefinition, we can send the relative angle to the robot instead of the absolute one according to the current situation of the robot. At the local server, we speak “go forward” or “turn right”, the local computer will recognize that and send the corresponding

Figure 10. Local UI.

Figure 11. Remote UI.

Table 2. Commands list.

Command	Brief description
Begin	start the robot drivers
Turn Left	turn left at a fixed angle
Turn Right	turn right at a fixed angle
Go Forward	go forward a fixed distance
Go Backward	go backward a fixed distance
Shake hands	lift its arm
Search	find some object

Table 3. User and robot dialog.

User	Robot
Begin	I'm ready for the commands
Turn left	Turn left 45^0
Turn Right	Turn right 45^0
Go Forward	Go forward one meter
Go Backward	Go backward one meter
Shake hands	Can I shake hands with you?
Stop	I will shut down

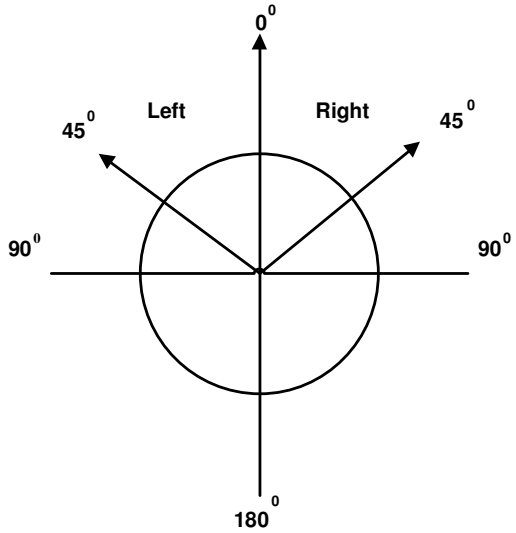

Figure 12. Angle coordinate system.

Table 4. Angel revolution.

right_angle_l=right_angle_l+45;
right_angle_r=right_angle_r+45;
turnright(right_angle_l,right_angle_r);
left_angle_l=left_angle_l-45;
left_angle_r=left_angle_r-45;
up_l=up_l+84.6;
up_r=up_r-84.6;
down_l=down_l-84.6;
down_r=down_r+84.6;

commands to the remote robot. When the robot receives that command, it will check the commands list. If the command is found, the related message will post to the diver routine, and the robot will act according to the user's will. Then, we could obtain the real-time positions of the robot, which is shown in Figures 13 and 14. From Figure 13, we can calculate the speed as 0.47936 m/s and 2.5% overshoot. Similarly, we can get the turn speed as 68.538° s^{-1}.Through these data, we can tell the velocity of the robot is much smaller than normal people, whose speed is approximately 1.5 m/s. This information is useful for analyzing the time delay of the transfer. As for the "stop" command, the robot will send "WM_CLOSE" message to shut down the drivers. While the basic speech commands (e.g. begin and turn right) are simple to execute, we will emphasize on the task "search". When we say the word "search", a dialog appears as shown in Figure 15. For example, we say "box", the speech is recognized and sent to the robot. The robot begins the corresponding procedure, as discussed earlier. The templates are shown in Figure 16. We try to include more templates and save them in the disc, and the commanded object is searched and matched until the object is found. First, we check the SURF algorithm in getting the initial position of the object using stationary images. Figure 17 shows the corresponding points between the object and the scene, while the corresponding position is drawn out in white line. Figure 18 shows the number of features it found both in the model and scene, as well as the matching time. From the two pictures, we can see that the SURF algorithm performs well in determining the position of the object, but we can also know that it consumes time very much. It spends 0.28 ms on extracting per feature, so we try to utilize camshift algorithm to realize the following tracking task, whose calculation time per frame is less than 10 ms. when camshift knows the initial position of the object, it can track it, as shown in Figure 19. The red rectangle marks the position. Figure 20 gives a comparison with using mean-shift, while we can see that the rectangle scale cannot change according to the distance between the object and the camera, which will make it difficult to track the object robustly.

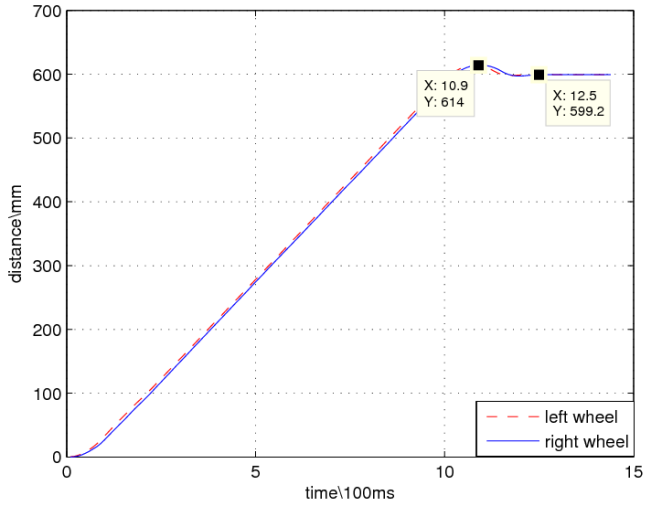

Figure 13. Go forward.

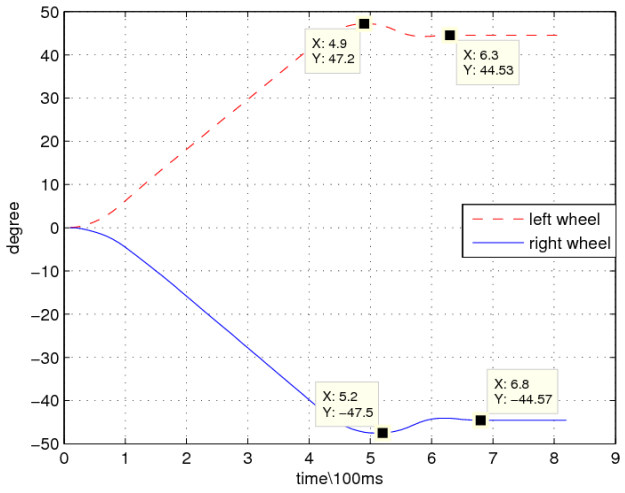

Figure 14. Turn right.

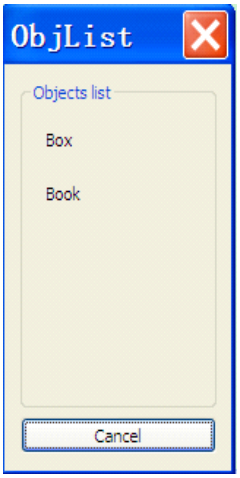

Figure 15. Object list.

Figure 16. Model list.

Figure 17. Corresponding.

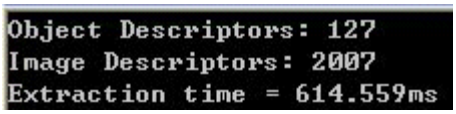

Figure 18. SURF features and time.

Figure 19. Camshift track.

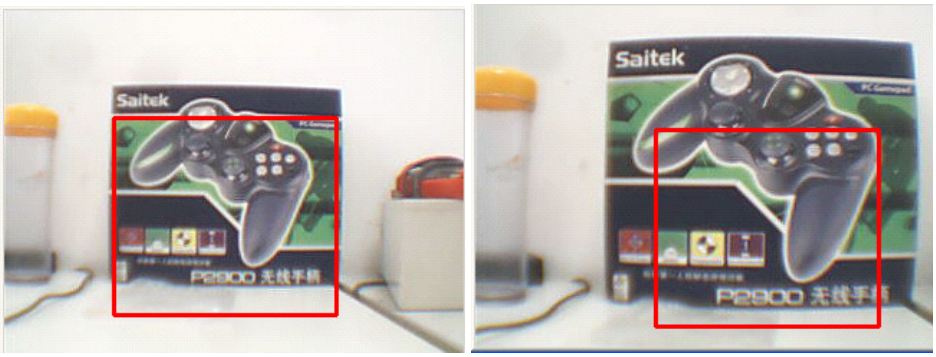

Figure 20. Mean-shift track.

Conclusions and future works

This paper presents a teleoperated speech controlled mobile robot structure. It incorporates teleoperation and voice control for a mobile robot. Voice control method makes the robot more intelligent and user-friendly. The event-based control method is incorporated and adjusted to handle the time delay, which makes the system robust. We implement the SURF and camshift to explore the environment, using the robot vision. Our experiments are used to verify the effectiveness of the proposed scheme. Now that we have a basic structure for the teleoperation control of the robot, we would like it to provide much more services and be more human-like, such as following and talking with people, exploring and understanding the surrounding and so on. It is a very promising and anticipated direction based on our current heuristic method.

ACKNOWLEDGEMENTS

This work is supported by Natural Science Foundation of China under Grant Nos. 60804003 and 60935001, and International Science & Technology Cooperation Program of China, No 0102011DFA10950.

REFERENCES

Albaker BM, Rahim NA (2011). Flight path PID controller for propeller-driven fixedwing unmanned aerial vehicles. Int. J. Phys. Sci., 6(8): 1947-1964.

Alessandri E, Gasparetto A, Valencia R, Martinez R (2005). An application of artificial intelligence to medical robotics. J. Intell. Robotic Syst., 41(4): 225–243.

Banga VK, Kumar R, Singh Y (2011). Fuzzy-genetic optimal control for robotic systems. Int.J. Phy. Sci., 6(2): 204-212.

Bay H, Ess A, Tuytelaars T, Gool LV (2008). SURF: Speeded Up Robust Features. Comput. Vision and Image Understand., 110(3): 346–359.

Bos J, Oka T (2007). A spoken language interface with a mobile robot. Artif Life Robotics. 11: 42–47.

Bradski GR (1998). Computer vision face tracking for use in a perceptual user interface. Intel Technol. J. 2nd Quarter.

Breazeal C, Brooks A, Gray J, Hoffman G, Kidd C, Lee H, Lieberman J, Lockerd A, Chilongo D (2004). Tutelage and collaboration for humanoid robots. Int. J. Humanoid Robotics, 1(2): 315–348.

Cielniak G, Duckett T, Lilienthal AJ (2010). Data association and occlusion handling for vision-based people tracking by mobile robots. Robotics and Autonomous Syst., 58: 435–443.

Fan B, Li K (2010). The speech control system of intelligent robot prosthesis. Intelligent Systems (GCIS), 2010 Second WRI Global Congress on. 2: 407-409.

Fong T, Scholtz J, Shah JA, Fluckiger L, Kunz C, Lees D, Schreiner J, Siegel M, Hiatt LM, Nourbakhsh I, Simmons R, Ambrose R, Burridge R, Antonishek B, Bugajska M, Schultz A, Trafton JG(2006). A preliminary study of peer-to-peer human-robot interaction. IEEE Int.Conference on Systems, Man and Cybernet., pp 3198 – 3203.

Foster ME, Giuliani M, Isard A, Matheson C, Oberlander J, Knoll A (2009). Evaluating description and reference strategies in a cooperative human-robot dialogue system. Int. Joint Conference on Artif. Intell., pp. 1818-1823.

John AB, Joana H, Robert R, Thora T (2010). A linguistic ontology of space for natural language processing. Artif. Intell., 174: 1027-1071.

Kang Y, Li Z, Shang W, Xi H (2010). Motion synchronization of bilateral teleoperation systems with mode-dependent time-varying communication delays. IET Control Theory & Appl., 4(10): 2129-2140.

Kim S, Gani MMM, Park SJ (2010). Uncertainty analysis of 3D motion data available from motion analysis system. Int. J. Phys. Sci., 5 (7): 1191-1199.

Li A, Wu D, M Z, Xu H (2009). Simultaneous sensor and hand-sensor calibration of a robot-based measurement system. Int. J. Phys. Sci., 4 (12): 846-852.

Li Z, Cao X, Ding N (2011). Adaptive fuzzy control for synchronization of nonlinear teleoperators with stochastic time-varying communication delays. IEEE Trans. Fuzzy Syst. In press.

Lu Y, Li L, Chen S, Huang Q (2010). Voice based control for humanoid teleoperation. Int. Confer. on Intell. Syst. Design and Eng. Appl., (ISDEA). pp 814 – 818.

Matthew ML, Nathan PK, Sonia HC, Chris VJ, Odest CJ (2009). Mobile human-robot teaming with environmental tolerance. Proceedings 4th ACM/IEEE int. conf. Human robot interact.

Mitsunaga N, Miyashita T, Ishiguro H, Kogure K, Hagita N (2006). Robovie-IV: A communication robot interacting with people daily in an office. IEEE/RSJ Int. Conference on Intell. Robots Syst., pp. 5066–5072.

Munir S (2001). Internet-based teleoperation. PhD Thesis, Georgia Institute of Technology.

Nakhaeinia D, Tang SH, Mohd Noor SB, Motlagh O (2011). A review of control architectures for autonomous navigation of mobile robots. Int. J.Phys. Sci., 6(2): 169-174.

Reinoso O, Fernandez C, Neco R (2007). User voice assistance tool for teleoperation. Advances in Telerobot., pp 107–120.

Schuller B, Can S, Feussner H, Wollmer M, Arisc D, Hornler B (2009). Speech control in surgery: A field analysis and strategies. Multimedia and Expo, 2009. ICME 2009. IEEE Int.Conference on. pp 1214-1217.

Takahashi M, Suzuki T, Shitamoto H, Moriguchi T, Yoshida K (2010). Developing a mobile robot for transport applications in the hospital domain. Robotics and Autonomous Syst., 58: 889– 899.

Thorission KR (1999). A mind model for multi-modal communicative creatures and humanoids. Int. J. Appl. Artif. Intell., 13(4-5): 449-486.

Veres SM, Lincoln NK (Sept 2008). Sliding mode control of autonomous spacecraft. Proc. TAROS'2008, Towards Autonomous Robotic Systems, Edinburgh. pp 1-3.

Xi N, Tarn T, Bejczy AK (1996). Intelligent planning and control for multirobot coordination: An event-based approach. IEEE Trans. Robotics Automat., 12(3): 439–452.