

An Visual System for Humanoid Robot Mobile-Manipulation Based on Virtual and Real Video Fusion

¹Lei Wang, ¹Lei Zhang, ¹Junyan Yi, ¹Xiaoyan Wen and ²Chuncheng Liao

¹*School of Electricity and Information Engineering, Beijing University of Civil
Engineering and Architecture, Beijing*

²*Department of Logistics Management, Beijing Normal University, Beijing*
E-mail: zhanglei@bucea.edu.cn

Abstract

A visual system based on virtual and real video for the humanoid robot manipulation is presented. 3D virtual model of humanoid robot is established. It has the same aspect and freedom setup as the real robot. Multiple feedback from the robot are fused and used to express the real robot status as text and images. The system also forecasts the operation order and displays the simulation result. In the data fusion module, a least-squares algorithm is adopted to calculate the real-time position and attitude of the robot. Experiments demonstrate that the system can offer good telepresence and a preview of the operation order. In this paper, we also propose an adaptive Elastic Net method for edge linking of images from the robot cameras to understand the situation of the task. In the proposed method, an adaptive dynamic parameter strategy and a stochastic noise strategy are introduced into the Elastic Net, which enables the network to have superior ability for escaping from local minima and converge sooner to optimal or near-optimal solutions

Keywords: *Humanoid robot, Elastic Net, Manipulation, Visual telepresence*

1. Introduction

For the humanoid robot manipulation, especially for mobile manipulation, it is important to get the whole visual scene of the robot and its worksite. One approach is to use the feedback real video images. Cameras were equipped in the remote robot. But the approach has the following disadvantages: the first, it is not applicable to the situation of poor visual conditions, for example, in the environment which is full of smoke. secondly, since the transmission of video images have a larger data, the larger network bandwidth needs to be occupied in the approach; the third, the operator can't observe the all detail of robot due to the limitation of perspective. On the other hand, extracting features from images is a common computer-vision problem. In particular, edge point images are problematic since there is inherently very little information available [1-3]. Such images consist of binary points on a two-dimensional plane which represent the edges and corners of an object. Then a feature-extraction algorithm should infer from the points a description of the object or objects present in the image. To understand the situation of the robot manipulation, the vision systems need to get the edge of the target. However, as a penalty method, the Elastic Net always suffers from a parameter tuning problem whenever applied to traveling salesman problems, of course a fundamental drawback of this type of networks [4].

In this work, we intend to enhance the ability of the Elastic Net to escape from local minima by introducing a stochastic noise strategy. Being different with other stochastic methods, the stochastic noise strategy enables the Elastic Net to obtain random dynamic properties from the problem states perspective. And the stochastic noise strategy is an

appropriate way to introduce stochastic characteristic. During the period of elastic net algorithm processing, edge points vibrate in their oscillation domains and therefore modify the neurons movement and matching relationship. With gradually decreasing the edge points' oscillations, the problem is reduced to the targeted edge linking problem. Thus the proposed algorithm is able to jump out of local minima and get convergence at a superior solution. Moreover by introducing an adaptive dynamic parameter strategy into the Elastic Net, we attempt to make the network have adaptive features so as to alleviate the problems of the Elastic Net mentioned above.

Another way to get the information of robot working is to use a virtual model of the robot work scene in the operator side. The virtual robot model has the same shape features and the DOF setting with the real robot and can receive a variety of the running data to express the state of real robot by the means of motion picture, meantime, establishing a predict simulation function module which has the same motion planning strategy with the real robot control system. In fact, these two kinds of video information are all important for humanoid robot mobile manipulation in real time as in teleoperation situation [5-8].

In the process of work, the various operating commands the operator send to robot can generate the robot execution process simulation data by the predict simulation module and are shown by the virtual scene. According to the predict result, operator can adjust the command strategy in time. The state data of robot received from the various types of sensors in the worksite are send to the operator side via the internet. After the fusion processing, generating the data can be received by the virtual scene and running the virtual robot mobile to achieve the real-time monitoring of robots. Due to the all kinds of data used by the virtual model are from the sensors in the worksite, its movement represents the true state of the robot.

At present, a lot of research institutions from many countries are conducting a similar study [9-11]. Using VR technology to reproduce the real time state of the robot is to overcome the problem for the real scene monitoring. The virtual interface used for monitoring of underwater robot provided by Qingping Lin [12] is representative. On the other side, only a few researchers have studied the humanoid robot teleoperation. The paper therefore develops the further study of the teleoperation-oriented virtual scene reconstruction technology on the humanoid robot BHR-02, constructed a practical application solved the calculation problem of matching and information fusion of multiple feedback. And achieved the humanoid robot teleoperation experiment using the system.

2. The Humanoid Robot Platform

The humanoid robot "BHR-02" (Figure 1) consists of ahead, two arms and legs, and has total 32 DOF (Degrees of Freedom), can perform complex movements flexibly such as walking, turning, up and down stairs, and martial arts [13-15]. The height is 160cm, and the weight is 63kg. The humanoid robot BHR-02 has a robot vision system that consists of two CCD camera and a voice system that consists of microphone and louder speaker device in the head. Each joint of robot has a DC motors as the driving force and the angle sensor to obtain the real-time angle of the joints. In addition, the robot has torque/force sensors at wrists and ankles, acceleration sensors and gyro sensors at the trunk. The feedback information of these additional sensor to control the stability of posture and pas of the robot in real time is very important. There are two computers built in robot body, one is for motion control, another for information processing (such as images processing, objects characters identifying and so on) and transferring data with remote cockpit. Using the dual port RAM to achieve the information interaction between two computers. When BHR-02 receives instructions from the remote cockpit, the first step of BHR-02 works is that one of its computers, which is for information processing and exchanging data, preprocesses the data and writes the results into the

dual port RAM. The second step is that the motion control computer reads the data from memolink, calculates and generates the values of motion trajectory that will be used to control corresponding DC motors. The control system of BHR-02 is a real-time position control system based on RT Linux operating system. On BHR-02 as an experiment platform, we have also done research on bipedal dynamic walking, 3D vision, motion planning, teleoperation and other subprojects.



Figure 1. The Humanoid Robot BHR-02

The operation process of the humanoid robot control system is planning the motion data (for example, the boards word *etc.*) of the robot off-line and saving the data files into the robot control computer. In the process, the control computer calls the corresponding gait data and adjusts the planning gait data in real time according to the real-time state information obtained by the various sensors to keep the stability of its posture and gait and the adaptability to the environment. So the actual operation process and the planning path is not exactly same. In addition, the robot control system can plan the motion trajectory of the upper limbs and the head according to the instructions to complete the upper limbs or head movement alone.

Remote cockpit used to control the humanoid robot BHR-02 is shown in Figure 2. whose system architecture is client/server mode based on the 802.11G wireless LAN. In client side, the operator can use composite input device to send out the control order. The composite input device includes keyboard, mouse, master arm and hand, joystick. The control order is received and parsed by the robot communication computer and the control computer receives and executes the order. In the server side, the information about the robot and its worksite can be collected by sensor system. Some of these data can be sent back to the operator side by the control computer through the communication compute. All the sensor system set in the robot worksite is used to get the motion information of the robot. Also the information from this system can be send back to the client side directly.

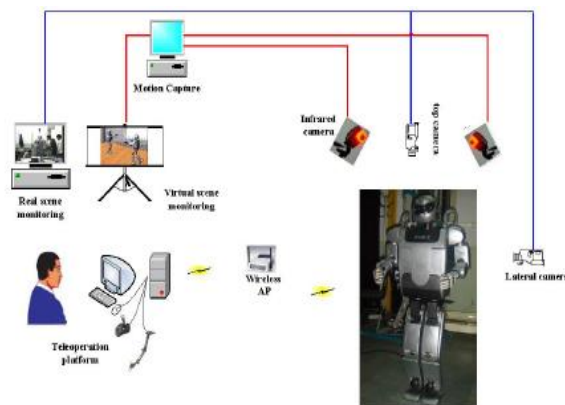


Figure 2. Working Sketch Map of the Remote Cockpit

The control orders send by the operator to the robot includes two kinds, one is the task space order, another is the joint space orders. The task space orders is refer to need to call the off-line data of the trajectory planning or calculate the trajectory data executed by each DOF online according to the task parameters. For example, the robot executes the fixed gait instructions, the robot executes the arm positioning operation *etc.* The joint space orders is that the robot, according to the instruction parameters, control the single or multiple joint motion directly. The kind of control instruction is mainly used to the arm pose tuning and head movement of the robot.

The feedback information of the robot to the operator site mainly includes the motion state information of the robot and the global position information. The robot control the computer to collect the data from the angle sensor equipped on each joint of the robot in every control cycle, these angle sensors data represents the current motion state of the robot. Besides, the all kinds of position sensors equipped on the robot worksite can detect the key point position coordinates on the robot body in real time. The position coordinates represents the real-time location of the robot in the current working space.

3. The System Constitution of the Telepresence System

In order to achieve the real-time state monitoring of the robot, the vision telepresence system based on the humanoid robot teleoperation system platform is built. The system structure of the vision telepresence system is shown in fig 3. Using the 3D modeling software to build the virtual robot model in the graphic workstation of the operation site. The model is similar with the real robot in the DOF configuration and the appearance characteristics. It has the same proportions of the whole body as the real robot. Meantime, the attribute matching module built for the robot model can match the motion data of the real robot to the model to drive the model to move. These data include the angle data of the each DOF of the whole body and the real-time position and attitude data of the robot relative to its work space. To the running robot model in 3D software, the operator can arbitrarily change the perspective and the display ratio to observe the macrostate and the detail of the movement. The operation control computer used by operator generate the various types of data needed for the robot Mobile motion in the virtual scene, these data through the network is transmitted to the graphics workstation. Two kinds of data is used to generate the driving data of the virtual robot model. One is the feedback data of the real-time from the sensors in the real robot worksite, the information generated by the fusion of these data will be used to reproduce the real robot movement. Another kind of data is made by predicting according to the order sent by the operator. these data is used to forecast and display the process of the robot executing commands, which is the representation of ideal motion process when robot executing the commands. Each kind of feedback from sensor is sent to the interface firstly and displayed as the text. Which include the robot real-time angle data of the each DOF, the robot real-time stability margin data, the force sensor and gyro sensor data and the distance parameters of the robot and target position, *etc.*

The robot real-time angle data of the each DOF of the feedback data from the field sensors can be directly forwarded to the virtual robot to drive model to show the robot relative motion of each joint. In order to accurately express the real-time position and posture of the robot in its operational space, system needs to obtain the corresponding parameters of real robot relative to the operation space. Real-time motion capture system can be used to get real-time motion information about any mark point in the space, recorded in the form of three-dimensional coordinate data. Installing such a landmark in robot is to get real-time motion data of the robot and transmit the three-dimensional coordinate data to the operator side through the network to calculate the real-time position and attitude data of the robot by the fusion. It is easy to know the position data, but the

three-dimensional coordinate data cannot be directly obtained. The fusion algorithm will be described in the section 3. The control commands from operator include the task space commands and the joint space commands. The commands are listed in the Table below.

Table 1. Teleoperation Operation Order

Order types	Order package length(byte)	Order information
Leg/whole body order	8	Motion type and motion parameter
Arm order	38	Whole 18 joints value of the two arm and hand
Head order	6	2 joints value of the head
Additional order	4	System initial control order type
Emergent stop order	2	Disposal of robot accident

The task space command refers to the task that the robot generates the motion trajectory, for example: leg movement, martial arts movements, *etc.* The data about the motion trajectory after offline planning is stored in the memory of the control computer in advance and is called by the control computer according to the different task. In the process of the order predicting of the system, the trajectory data will be stored on the computer in the form of files. System will select the corresponding trajectory data accordance with the different type of commands. These data include all the information needed to build the virtual model. The joint space instruction is that the actual joint angle of each working joint is included in the control instructions. System will calculate to generate the predicted data about the type of order directly according the order parameters.

All the operator order prediction is based on the current state command to execute the prediction. So the system reads the real-time state data of the robot at this time firstly, that is to according to the feedback about the current state from all the sensors to execute the prediction.

4. The Calculation Method for the Robot Real-time Pose

As mentioned, in the process of the robot movement, the real-time angle data of each joint DOF can be obtained by each robot sensor system of the robot. These angles data can be used to drive robot model to express the relative movement among the various parts of the robot. In addition, in order to make the virtual robot model show its real state accurately, not only to use the angle data, but also to express the position and pose of the robot relative to its working space. Using the motion capture system, the three-dimensional position coordinate data of each mark point of the robot body relative to its working space can be captured. These coordinate data can be used to seek the real-time attitude changing parameter of the robot body [13-15].

As $P_1, P_2, P_3 \dots P_m$ are marks attached in the robot body. The position vector of the marks at time $t=t_1$ and $t=t_2$ is denoted by a_i and p_i . The position coordinate of the centroid point p_0 is a and p.

As:

$$a = \frac{1}{m} \sum_{i=1}^m a_i \quad (1)$$

$$p = \frac{1}{m} \sum_{i=1}^m p_i \quad (2)$$

In addition, by each marked point relative to the centroid vector $a_i - a$ is used to describe the various signs, it can be defined as the distribution matrix A :

$$A = \frac{1}{m} \sum_{i=1}^m (a_i - a)(a_i - a)^T \quad (3)$$

At the moment t_2 the matrix:

$$P = \frac{1}{m} \sum_{i=1}^m (p_i - p)(p_i - p)^T \quad (4)$$

Without considering the various signs system error of position coordinate measuring point, and the flexible change and robot rigid conditions, the robot pose calculation of the trunk part. The movement from t_1 to t_2 can be expressed by translation vector r and rotation matrix R of P_0 the center of mass of three marks. The rotation matrix R should satisfy:

$$R^T R = I; \det(R) = +1 \quad (5)$$

According to the Euclidean transformation, for any one of point can be expressed in the following Equation:

$$p_i = a + r + R(a_i - a) \quad (6)$$

In order to obtain the translation vector r and rotation matrix R above, the least-squares which is not weighted should be considered. Assume that the r and R approximation have been obtained h and H . the h and H should make the least squares function type has a minimum value:

$$f(h, H) = \frac{1}{m} \sum_{i=1}^m \left[(p_i - a - h - H(a_i - a))^T \times (p_i - a - h - H(a_i - a)) \right] \quad (7)$$

The $p_i - a - h - H(a_i - a)$ in the Equation Is each symbol vector point of actual measurement, and Difference between using The results of calculation of h and H . The smaller the value of f , The calculation results will be more accurate. Equation can be obtained by transformation:

$$f(h, H) = (p - a - h)^T (p - a - h) + \text{tr}(P - 2GH^T + HAH^T) \quad (8)$$

This type of matrix trace calculation, Expressions G is defined as follows:

$$G = \frac{1}{m} \sum_{i=1}^m (p_i - p)(a_i - a)^T \quad (9)$$

f reached the minimum condition is for variables h and H obtain infinitesimal Δh and ΔH . The amount of change of f is $\Delta f = f(h + \Delta h, H + \Delta H) - f(h, H)$ must be 0. For variables h is $h = p - a$, For variables H is $HA = G$. This can be obtained in (6) type of the variable is:

$$r = p - a \quad (10)$$

$$R = GA^{-1} \quad (11)$$

Further R can be obtained by spatial rotation angle of robot is part of the trunk:

$$\begin{cases} \beta = A \tan 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\ \alpha = A \tan 2(r_{21}, r_{11}) \\ \gamma = A \tan 2(r_{32}, r_{33}) \end{cases} \quad (12)$$

The type of α , β , γ is the rotation angle in the three axis robot from the trunk part time. So the type (10) and (12) changes in trunk space robot pose parameters expression

5. The Adaptive Elastic Net Method for Edge Linking

As mentioned above, the Elastic Net algorithm is geometric in nature and appears to be more suitable for solving the edge linking problems, but it still suffers from three major problems: the tough parameter tuning, relatively slow running time and solution quality problem. All these problems have not been satisfactorily solved so far. In order to improve its convergence speed and solution quality for the edge linking, and alleviate the parameter tuning problem, we should make the network have adaptive features. Therefore an adaptive Elastic Net method is proposed by introducing an adaptive dynamic parameter strategy and a stochastic noise strategy into the original Elastic Net, as defined below:

$$E = -\alpha(t)K \sum_{i=1}^n \ln \sum_{j=1}^m e^{-|x_i' - y_j|/2k^2} + \beta(t) \sum_{j=1}^m |y_j - y_{j-1}| \quad (13)$$

The positions of the points defining the rubber band are updated according to the formula:

$$\Delta y_j = \alpha(t) \sum_i w_{ij} (x_i' - y_j) + \beta(t) K (y_{j+1} - 2y_j + y_{j-1}) \quad (14)$$

where:

$$\begin{cases} \alpha(t+1) = (1 - \varphi)\alpha(t) & \text{if } \alpha(t) > \alpha_{\min}, \\ \alpha(t+1) = \alpha_{\min} & \text{otherwise.} \end{cases} \quad (15)$$

$$\begin{cases} \beta(t+1) = (1 + \eta)\beta(t) & \text{if } \beta(t) < \beta_{\max}, \\ \beta(t+1) = \beta_{\max} & \text{otherwise.} \end{cases} \quad (16)$$

$$x_i' = x_i + \frac{\delta(t)}{2} \text{rad}(-d_i, d_i) \quad (17)$$

$$\delta(t+1) = (1 - \lambda)\delta(t) \quad (18)$$

Here φ and η are small positive constants, which are selected empirically. d_i is the distance between edge point i and its nearest neighbor edge point. λ is the damping factor of the relative stochastic noise ($0 < \lambda \leq 1$)

Eqn. (15) and Eqn. (16) are responding to the adaptive dynamic parameter strategy, which provides a flexible method of tuning parameters to avoid elastic net being saturated at non-feasible solutions. Eqn. (17) and Eqn. (18) are responding to the stochastic noise strategy, which contains the decaying control to reduce oscillations of the edge points to be the targeted problem.

In the improved algorithm the stochastic noise strategy reveals that annealing stochastic noise on problem states is also an effective way to improve performance of deterministic algorithm. The stochastic noises embedded on confirming edge point positions could cause oscillations of the edge points, and further influence the force that moves a neuron on the rubber band towards those edge points. With the decrease of the controlling parameter $\delta(t)$, stochastic noises gradually disappear and edge points oscillations are terminated, the edge points are finally located on their original positions. Due to the limitation of distance d_i , the position oscillations are controlled in local domain of every edge point. The improved Elastic Net algorithm not only obtains local stochastic searching ability, but also has superior ability for escaping from local minima. It is important to note that without the limitation of d_i neurons will be confused utterly to break away the restraint of searching shortest tour length, then the solutions are bad predictably.

In the proposed method constant parameters α and β are replaced with two variables $\alpha(t)$, $\beta(t)$ in Eqn. (13) and (14). And $\alpha(t)$, $\beta(t)$ are updated according to (15) and (16)

respectively. In the proposed model, the values of $\alpha(t)$ and $\beta(t)$ must be kept in bounds, and α_{\min} and β_{\max} are the bounds assigned to $\alpha(t)$ and $\beta(t)$. Based on the analysis of various parameters involved in the Elastic Net [16] and our experiments, in this paper: $\alpha_{\min} = 0.1$, $\beta_{\max} = 6.0$.

In the original Elastic Net, α represents the influence of the first kind of force which moves a unit on the path towards those edge points to which it is nearest. β represents the influence of the second type of force, which pulls a unit towards its neighbors on the path to minimize the total path length. Hereby constants α and β could balance the relative strength between these two types of forces. If α is relatively big or β is relatively small, the network could find all edge points in shorter time, but crossovers are easy to be formed on the path. Crossovers could increase the total contour length and make the solution non-optimal. On the contrary, if α is relatively small or β is relatively big, it will take a long time for the rubber band to be extended, and more importantly in feasible contours caused by missing edge points would be produced.

As is well known, the original Elastic Net algorithm tracks to a local minimum of energy function when $K \rightarrow 0$. It is desirable that this minimum that is tracked to remains the global minimum. Unfortunately, this cannot be guaranteed and many local minima exist [17]. In the proposed model we have replaced constants α and β with variables $\alpha(t)$ and $\beta(t)$, which makes the Elastic Net have adaptive features. Below we will give the analysis. Initially, $\alpha(t)$ is set as a relatively large value, $\beta(t)$ is initialized as a relatively small value. Then the first item of the energy function $-\alpha(t)K \sum_{i=1}^n \ln \sum_{j=1}^m e^{-|x_i - y_j|/2k^2}$ in Eqn. (13) would play a key role in the network at the early stage. This means at the beginning the first kind of force that moves net units towards those edge points is very strong, and the second kind of force that pulls net units towards their neighbors on the path is weak. Under strong attraction by the edge points, it will take shorter time for the units move near to the edge points. Thus the extending process of the network is accelerated. Furthermore, as the attraction of net units by each edge point is too strong to be neglected, and the force among neighboring units is very weak, every edge point would be surrounded by some net units and no one would be missed. Therefore the network's ability of searching for edge points is increased, and related local minima as well as in feasible solutions result from missing edge points could be prevented effectively.

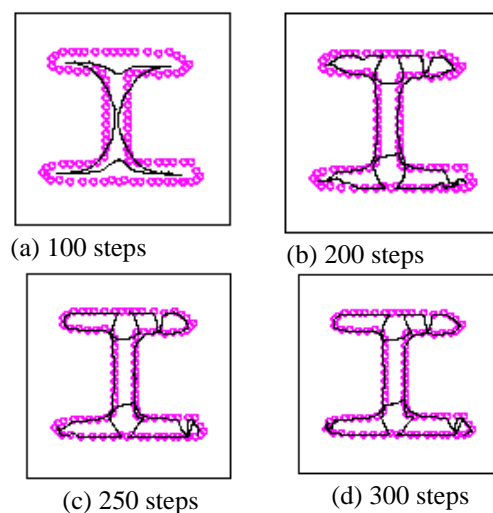


Figure 4. Evolution of Edge Linking Contour for Image of an I Shape using the Original Elastic Net Approach

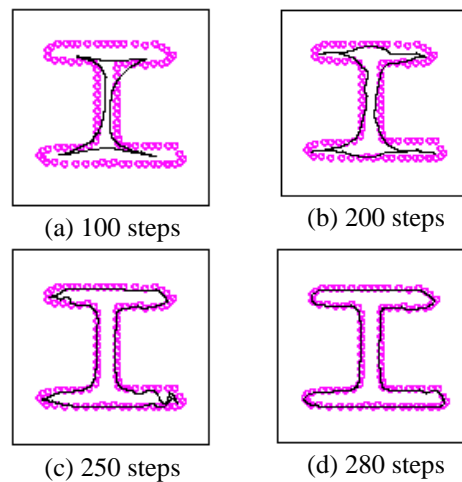


Figure 5. Evolution of Edge Linking Contour for Image of an I Shape using the Proposed Method

As time goes by, $\alpha(t)$ is decreased and $\beta(t)$ is increased gradually according to Eqn. (15) and (16). When $\beta(t)$ becomes sufficiently large, $\alpha(t)$ reaches a relatively small value, the second item $\beta(t)\sum_{j=1}^m |y_j - y_{j+1}|$ of the energy function in Eqn. (13) would play an important role. Then the second type of force that pulls units towards their neighbors on the path becomes stronger. Strong attraction by their path neighbors makes the net units close with one another, which helps the network to form the shortest contour and accelerates the convergence process. Thus, non-optimal local minima are prevented, and optimal or near-optimal solutions could be obtained in shorter time.

Figure 4 and Figure 5 show the evolutions of the edge linking contour of the original Elastic Net approach and the proposed method for the same edge linking problem of an I shape image.

6. Experiment

In this research, we use Maya to achieve the modeling and state display. Through making the skin of the virtual character and skeletal model respectively and binding them, Maya can render the arbitrary motion of the virtual role. In addition, by designing the function plug-in of Maya, the function of data processing can be developed easily. The functional plug-in in this topic developed mainly completes the reading of the data structure and the matching of data and model.

The use of multithreading technology, we design a teleoperation client (*i.e.*, the operator end) control platform program. The data communication part uses the wireless local area network 802.11g, the SOCKET programming based on TCP/IP achieved the bidirectional reliable transmission between the client and server. Including the bidirectional reliable transmission the bidirectional reliable transmission between the platform control computer and the robot communication sever computer, the motion capture sever and the virtual scene.

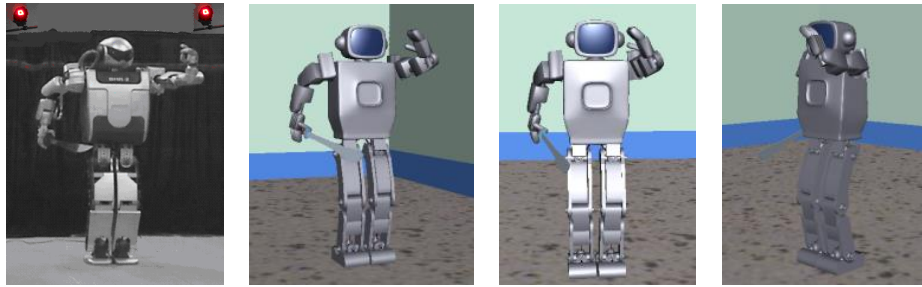


Figure 6. Sabreplay and Virtual Scene

The robot motion control computer in each control cycle, complete the motion data acquisition and send to the operator side. By capture of real time motion robot, we use the digital optical system based on infrared capture, its accuracy can reach 0.1 mm, and the sampling frequency can reach up to 60 frames/sec. The time controller Maya interface in the virtual scene which refresh rate of 24 frames/sec. All kinds of data transmitted to the operator side will be recorded as data file and saved in the client computer. When completed the operation, the data can be called again to browse historical process of the robot.

The work process of the robot and the results by the system described in this article is shown in Figure6. The operator can change the viewpoint and the scaling of the zoom freely.

7. Conclusion

For the purpose of remotely controlling humanoid robot BHR-02 manipulation, we designed a visual system. The system has the following characters:

(1) The operator can change the viewpoints and the scaling of the zoom freely. Therefore the operator can observe the detail of the models. Combining with the text display of all the data, operator can have a real feel.

(2) The virtual model can receive and render the angle data of the real-time robot motion and the data from the data fusion of the motion capture system. The fusion algorithm makes the capture of the robot real-time position and attitude not to use the special system.

(3)The virtual model can receive the order prediction simulation data and historical data to display, which offer the prediction to the order and the browsing of history data to operate.

(4)An algorithm for adaptive elastic net method for edge linking is presented.

(5)The system shown in the article in the BHR-02 humanoid robot is provided.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61473027), sponsored by Ministry of Housing and Urban-Rural Development of The People's Republic of China, Project's "The Research of Power Quality Monitoring and Energy Saving for Intelligent Building with Multiple Equipment" (2012-K1-42); also sponsored by BUCEA Urban Rural Construction and Management Industry Research Development Collaboration Post Graduate Training Centre (cxy2014175)

References

- [1] S. J. Gilson and R. I. Damper, "An empirical comparison of neural techniques for edge linking of images", *Neural Computing and Applications*, vol. 6, no. 2, (1997), pp. 64-78.
- [2] Q. Huang, K. Li and T. Wang, "Control and Mechanical Design of Humanoid Robot BHR-01", *Proceeding of the Third IARP International Workshop on Humanoid and Human Friendly Robotics*, vol. 12, (2002), pp. 10-13.

- [3] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie, "Planning Walking Pattern for a Biped Robot", IEEE Transactions on Robotics and Automation, vol. 17, no. 3, (2001), pp 280-289.
- [4] J. V. Stone, "The optimal Elastic Net: Finding solutions for the traveling salesman", Proceeding of International Conference on Artificial Neural Networks, Brighton, England, (1992), pp. 170-174.
- [5] G. Burdea, "Invited Review: The Synergy between Virtual Reality and Robotics", IEEE Transactions on Robotics and Automation, vol. 15, no. 3, (1999) June, pp. 400 – 410.
- [6] Monferrer, Alexandre, Bonyuet, and David, "Cooperative Robot Teleoperation through Virtual Reality Interfaces", IEEE Proceedings of the Sixth International Conference on Information Visualisation (IV'02), 1093-9547/02.
- [7] R. Igor, R. C. Belousov and G. J. Clapworthy, "Virtual Reality Tools for Internet Robotics", Proceedings of the 2001 IEEE International Conference on Robotics 8, Automation Seoul, (2001) May 21-26, pp. 1878-1883.
- [8] Monferrer, Alexandre, Bonyuet, and David, "Cooperative Robot Teleoperation through Virtual Reality Interfaces", IEEE Proceedings of the Sixth International Conference on Information Visualisation (IV'02), 1093-9547/02.
- [9] G. Burdea, "Invited Review: The Synergy between Virtual Reality and Robotics", IEEE Transactions on Robotics and Automation, vol. 15, no. 3, (1999) June, pp. 400 – 410.
- [10] S. Kagami, J. J. Kuffner Jr., K. Nishiwaki, T. Sugihara, T. Michikata, T. A. M. Inaba and H. Inoue, "Design and Implementation of Remotely Operation Interface for Humanoid Robot", Proceedings of the 2001 IEEE International Conference on Robotics & Automation, (2001) May 21-26, pp. 401-406.
- [11] S. Kagami, J. J. Kuffner Jr., K. Nishiwaki, T. Sugihara, T. Michikata, T. A. M. Inaba and H. Inoue, "Design and Implementation of Remotely Operation Interface for Humanoid Robot", Proceedings of the 2001 IEEE International Conference on Robotics & Automation, (2001) May 21-26, pp. 401-406.
- [12] Q. Lin, C. Kuo, Frina, Fsut and Frse, "Virtual Tele-Operation of Underwater Robots", Proceedings of the 1997 IEEE International Conference on Robotics and Automation Albuquerque, (1997) April, pp. 1022-1027.
- [13] L. Zhang, Q. Huang, Y. Lu, T. Xiao and J. Yang, "A Visual Tele-operation System for the Humanoid Robot BHR-02", Proceedings of the 2006 IEEE International Conference on Robotics and Intelligent System, (2006), pp. 1110-1116.
- [14] L. Zhang, Z. Jiang, Q. Huang, Y. Lu and T. Xiao, "A reconstruction method for working scene of humanoid robot BHR-02 teleoperation based on multiple information fusion", Proceedings of the IEEE International Conference on Robotics and Biomimetics, (2007).
- [15] Y. Lu, Q. Huang, M. Li, X. Jiang and Q. Xu, "3D-Simulation for the Teleoperation of the Humanoid Robot BHR-02", Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China, (2008) September.
- [16] A. I. Vakhutinsky and B. L. Golden, "The co-adaptive neural network approach to the Euclidean Traveling Salesman Problem", Neural Networks, vol. 16, no. 10, (2003), pp. 1499-1525.
- [17] J. Wang, Z. Tang, Q. Cao and X. Xu, "An elastic net learning algorithm for edge linking of images", IEICE Trans. Fundamentals, vol. E86-A, no. 11, (2003), pp. 2879-2886.

