Robot Motion Control and Image Reconstruction over Internet

C.D. Cheng, P. Vadakkepat*, C.C. Ko, B.M. Chen, and X. Xiang
Department of Electrical and Computer Engineering
National University of Singapore, Singapore 117576
Email: eleccd, prahlad@nus.edu.sg

Abstract

An image reconstructing scheme for video feedback over the network for cyber robot motion control is proposed. The suggested system has better real time update characteristic, and has more practical feel for the remote user. For the cyber robot control via internet, the real time video feedback is necessary. However, this case is not always practical due to network congestion and the associated delays. In the proposed scheme, the motion image of the robot is not required to transfer constantly to the remote user via the network. Instead, the sampled position and orientation of the robot are send to the client, and the image is reconstructed. This helps to save a large amount of network bandwidth. The image can be updated as fast as in local mode. Furthermore, the user can have the same real feel as the actual image feedback. The reconstructing algorithm in the scheme needs additional computation and storage space. The pixel color values also to be archived in advance. Irrespective of all these, the performance is still found to be much better than the direct image feedback over network.

Keywords: Cyber Robot; Image Reconstruction; Remote Control over network.

1 Introduction

Remote access of experimental setup has gained attention recently [1, 2, 3, 7, 10]. In remote control of experiments, the real time video feedback is necessary in order to facilitate real-time control. Owning to the limited network bandwidth resources, network congestion produces delay. Therefore, the real time image feedback is generally difficult, especially in the public networks such as Internet. When the network delay is larger, obtaining real time video feedback is almost impossible. If the remote user is not able to receive the image feedback within a specified time, the work performance will tend to degrade. Simulation methods are useful to act as an alternative to the information feedback introduced in [8, 9, 10, 11].

In the web-based robot experiment developed at Department of Electrical and Computer Engineering of the National University of Singapore, students can conduct the robot control experiments via the Internet with the real time video feedback. The whole experiment structure is depicted in Figure 1. In this experiment, a robot control program runs on the program server located in the laboratory. This server receives the request from the remote user and controls the mobile robot. The position and orientation angle values of the robot are sent to the client. The video server is used to provide the real time video source and deliver it to the client. Web server provides the authorization service as the system allows one user at a time to experiment with the robot. The server verifies the user name and password, and allows a single user to log on the system in order to avoid conflicts. The experiment is aimed to help the students to familiarize with the robot control principles and methodology, and to let them conduct the experiment from anywhere.

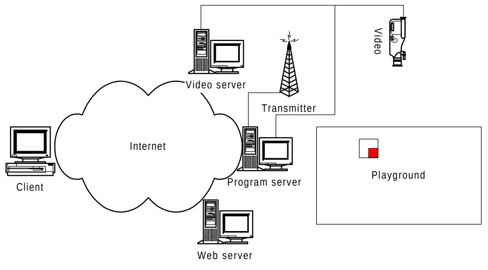

Figure 1: Web-based robot experiment structure

In general, the cyber robot experiment works very well within the Intranet of the National University of Singapore. However, the performance was found to degrade outside the NUS intranet, because of the narrow network bandwidth. Although the simulation methods can be resorted to, it is not possible to provide the actual feel of the experiment to a user. To tackle this, an efficient scheme to improve the feedback performance in narrow network bandwidth solution is implemented. In this scheme, each frame of the video is driven by the position (x and y coordinate values) and orientation angle data of the robot rather than its image data. Accordingly, the amount of data to be transmitted over the network greatly reduces, and it can produce faster image update. This scheme takes advantage of the image transformation techniques, including coordinate shift and rotation, to reconstruct the image.

In fact, for some of the popular video servers such as Inetcam and netmeeting, the image is being constantly updated even in case of a stationary robot. This case results from the variations in surroundings such as the changes in light. When the robot is in motion, the amount of data transmitted over the network is larger. Consequently, conducting this experiment by real time video feedback over Internet is impractical.

So far, most of the video sources are from the video servers that adopt the image compression techniques such as JPEG, MPEG, etc. These techniques are meant for general purposes. However, these techniques are not suitable for a rigid object such as the cyber robot. This type of object has a rigid shape in nature, and the relative positions of the image pixels are fixed. Thus, After obtaining the image attributes of the object, the subsequent image motion can be characterized by the x coordinate, y coordinate and the orientation angle values. This is because the 2-D image of this kind of object can be completely represented by the above three degrees of freedom. However, the pixel values of the image has to be acquired prior to the image reconstruction. In the following, we take advantage of this attribute to characterize the robot motion.

In this paper, an image reconstructing scheme over network for the cyber robot motion is presented. After receiving the pixel color values of the playground and robot, the subsequent image update is made by the position and orientation angle values. In the course of image reconstruction, the coordinate rotation and shifting are performed. This approach can meet the fast image update rate and provide a real feel. The entire computation time and storage space required to update the image are acceptable. This algorithm requires a small amount of data for video feedback over Internet, and it has the advantage of saving in the network bandwidth.

The remainder of this paper is organized as follows. After the introduction and the web-based robot experiment development, the proposed image reconstructing algorithm is presented in Section 3. Section 4 provides a detailed performance analysis on the computation, storage space and data transmission of the proposed algorithm, and compares it with the existing schemes. The experimental results for the web-based robot experiment and a comparative study with the Inetcam video server are also presented in this Section. The results show that the proposed method outperforms the existing methods. Finally, conclusions are drawn in Section 5.

Throughout this paper, the following notations are used. Given a playground image with $m \times n$ pixels and robot image with $l \times l$ pixels,

$(X,\ Y)$:= denotes the coordinate value in robot coordinate system.

$(x,\ y)$:= denotes the coordinate value in playground coordinate system.

$r(X,\ Y)$:= denotes the pixel color value of coordinate $(X,\ Y)$ in robot coordinate system.

$g(x,\ y)$:= denotes the pixel color value of coordinate $(x,\ y)$ in playground coordinate system.

$\boldsymbol{S}^r$:= denotes the coordinate collection containing all pixels in robot coordinate system, and there exists $\boldsymbol{S}^r = \{(X, Y)\}$.

$\boldsymbol{S}_i^g$:= denotes the coordinate collection of pixels covered by the robot in the playground coordinate system in the ith frame, and there exists $\boldsymbol{S}_i^g = \{(x, y)\}$.

C^r := denotes the $l \times l$ matrix reflecting the robot pixel colors.

C^g := denotes the $m \times n$ matrix reflecting the playground pixel colors.

α := the orientation angle of robot to playground coordinate system, which varies from 0° to 360°.

$(X_0,\ Y_0)$:= the coordinate of the center of the robot in the playground coordinate system.

Moreover, for a given set $\boldsymbol{X}$, $|\boldsymbol{X}|$ denotes the total number of elements in $\boldsymbol{X}$.

2 Web-based Robot Experiment

Before introducing the image reconstructing algorithm, the web-based robot experiment is described. The experiment is part of a course for the final-year students of the Department of Electrical and Computer Engineering of the National University of Singapore. The students can conduct this experiment through Internet "www" technology. In the experimental setup, the robot control system consists of a CCD camera, a mobile robot, playground, RF transmitter and the software running on the computer, as shown in Figure 1. The experimental system is primarily used for robot soccer competition. The NUS team has participated in several international robot soccer competitions. The robot soccer setup is used to teach the students the principle and methodology of robot control. Several robot control algorithms have been developed for the students to test, and the students are allowed to modify some of the parameters in these algorithms. The experiment is aimed to help the students to receive a deeper impression of robot control. In addition, this experiment is also useful to test the performance of the developed algorithms.

As showed in Figure 1, a CCD camera with a resolution of 240 × 360 pixels is used to capture the image of robot and the playground. The captured image is sent to the video card installed in the computer for further processing. To differentiate the robot and the playground easily, the robot is painted in different color from the surrounding. The program on the computer computes the position and orientation of the robot by analyzing the color of the pixels in the image. After computing the position and orientation, the program makes a decision to drive the robot according to the control algorithms mentioned above. The robot is commanded through an RF transmitter. The transmitter is connected to the computer through RS232 serial line, and the command obeys the specific transmission protocol so that more robots can be controlled through the same transmitter and the same wireless frequency.

Since the experiment can be conducted over Internet, the TCP/IP connection is established between the client and the server. The connection transmits the user's command to the program server, and return the feedback information such as the position and orientation of the robot to the client. However, it is required to build another connection to transmit the real time video stream to the client. The video server can transmit the real time video to the client via a broadband network. For narrow-band transmission, the real time image feedback can not guaranteed. The image reconstruction method is studied instead in order to improve the vision feedback.

3 Image Reconstructing Algorithm

A. The Coordinate System Representation

In this section, the robot and playground coordinate representations are presented. The robot image with 7 × 7 pixels is considered. Figure 2 shows the definition of the robot coordinate system, where the center of robot is defined as origin.

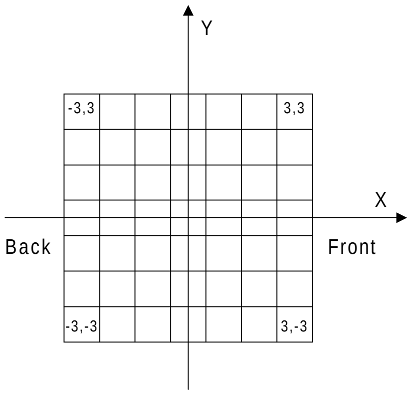

Figure 2: Robot pixel coordinate representation

The coordinate set $\boldsymbol{S}^r$ is as shown in Example 3.1,

Example 3.1 Suppose $l = 7$, then $\boldsymbol{S}^r = \{(-3,\ +3),\ (-3,\ +2),\ (-3,\ +1),\ (-3,\ 0),\ (-3,\ -1),\ (-3,\ -2),\ (-3,\ -3);\ (-2,\ +3),\ (-2,\ +2),\ (-2,\ +1),\ (-2,\ 0),\ (-2,\ -1),\ (-2,\ -2),\ (-2,\ -3);\ (-1,\ +3),\ (-1,\ +2),$

$(-1,\ +1)$, $(-1,\ 0)$, $(-1,\ -1)$, $(-1,\ -2)$, $(-1,\ -3)$; $(0,\ +3)$, $(0,\ +2)$, $(0,\ +1)$, $(0,\ 0)$, $(0,\ -1)$, $(0,\ -2)$, $(0,\ -3)$; $(+1,\ +3)$, $(+1,\ +2)$, $(+1,\ +1)$, $(+1,\ 0)$, $(+1,\ -1)$, $(+1,\ -2)$, $(+1,\ -3)$; $(+2,\ +3)$, $(+2,\ +2)$, $(+2,\ +1)$, $(+2,\ 0)$, $(+2,\ -1)$, $(+2,\ -2)$, $(+2,\ -3)$; $(+3,\ +3)$, $(+3,\ +2)$, $(+3,\ +1)$, $(+3,\ 0)$, $(+3,\ -1)$, $(+3,\ -2)$, $(+3,\ -3)\}$.

The coordinate system of the playground has the left-bottom corner functions as the origin of the coordinate system (Figure 3).

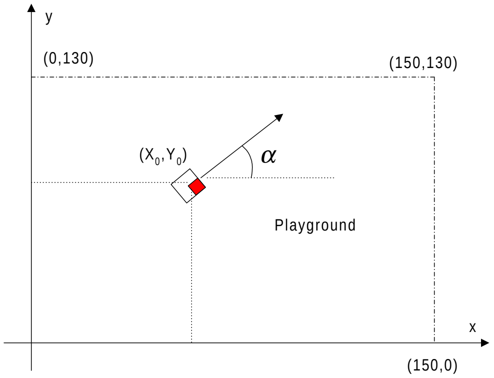

Figure 3: The playground coordinate system

Since a playground with 150 centimeters in length and 130 centimeters in width is used, the range of the horizontal coordinate of image is defined from 0 to 150, and the vertical coordinate is defined from 0 to 130.

B. Coordinate Conversion for Pixels

The following procedure of the coordinate conversion is constructed according to Equation 1, which is fundamental to the algorithm implementation. The relationship between the two coordinate systems is illustrated in Figure 3.

$$\begin{cases} x = X \cdot \cos\alpha - Y \cdot \sin\alpha + X_0 \\ y = X \cdot \sin\alpha + Y \cdot \cos\alpha + Y_0 \end{cases} \tag{1}$$

According to 1, after getting the position (X_0, Y_0) and orientation angle α, theoretically a unique map from $\boldsymbol{S}^r$ to $\boldsymbol{S}^g_i$ in the ith frame image can be obtained.

$$\boldsymbol{S}^r \rightharpoonup \boldsymbol{S}^g_i \qquad i = 1, 2, 3, \cdots \tag{2}$$

Obviously, there exists $|\boldsymbol{S}^g_1| = |\boldsymbol{S}^r|$.

Example 3.2 Suppose $X_0 = 57, Y_0 = 45$ and $\alpha = 120$, then by using 2, the coordinate $X = 1, Y = 3$ is converted to $x = 53, y = 44$.

C. Image reconstruction

Algorithm 3.1 Image reconstruction of robot motion over Internet.

Step 0. Extract all the playground pixel color values and the robot pixel color values over network, and store them in C^r and C^g, respectively. After receiving the position and orientation angle values from the server, the coordinates (X, Y), $(X, Y) \in \boldsymbol{S}^r$ of the robot coordinate system are converted to the playground coordinates (x, y), $(x, y) \in \boldsymbol{S}^g_1$. The conversion is carried out one element at a time, and (x, y) is redrawn using the corresponding pixel color value $r(X,\ Y)$ till all the elements of $\boldsymbol{S}^r$ are completed.

Step 1. Let $i := i + 1$, receive the new position and orientation angle values from the network. If the position and orientation angle values in the $(i+1)$th frame are the same as those in the i frame, move to the beginning of Step 1, otherwise go to Step 2.

Step 2. Convert the coordinate (X, Y), $(X, Y) \in \boldsymbol{S}^r$ in the robot coordinate system to the coordinate (x, y), $(x, y) \in \boldsymbol{S}^g_{i+1}$ in the playground coordinate system one element at a time, and redraw (x, y) using the corresponding pixel color value $r(X,\ Y)$ till all the elements of $\boldsymbol{S}^r$ are finished. When $(x, y) \in (\boldsymbol{S}^g_i - \boldsymbol{S}^g_i \bigcap \boldsymbol{S}^g_{i+1})$ holds, redraw (x, y) using the pixel color value $g(X,\ Y)$ till all elements of the set $(\boldsymbol{S}^g_i - \boldsymbol{S}^g_i \bigcap \boldsymbol{S}^g_{i+1})$ are done. Go to Step 1.

⋆

In this implementation, the image of the playground pixel color values and the robot pixel color values are required by the server prior to the implementation of Algorithm 3.1 so that subsequent frames can be reconstructed by the position and orientation angle values.

To illustrate the Algorithm 3.1, the source code of the program in Java language is made available. The complete Java code is available for download from http://vlab.ee.nus.edu.sg/ccd/download/.

In this program, the function *robotconversion* is for the coordinate conversion, and function *drawrobot* is to redraw the image.

In Algorithm 3.1, to facilitate the search of the pixel color values in C_g and C_r, it is required to further convert the above two coordinate systems to the Java program coordinate system. The origin of the playground coordinate system is considered at the left-bottom corner of normal coordinate system. However the origin is at the left-top corner in the Java program coordinate system. In addition, it is required to consider the image object position in the Applet. The playground pixel color values can be searched with the following equations.

$$C_g[x + left][130 - y + top], \tag{3}$$

where $left$ implies the x coordinate of upper-left corner of the image, and *top* implies the y coordinate of upper-left corner of the image. In the program, $left = 30$, and $top = 10$.

Meanwhile, the robot pixel color values is provided by the following equation,

$$C_r[X + 3][Y + 3] \tag{4}$$

4 Performance analysis

The data to be transmitted for reconstructing image over network are only the position and orientation angle. The computation time and storage space required of the proposed algorithm is analyzed in this section. From the Algorithm 3.1, the number of pixels needed to update per frame η_1 is

$$\eta_1 = |\boldsymbol{S}_i^g| + |\boldsymbol{S}_{i+1}^g| - |\boldsymbol{S}_i^g \bigcap \boldsymbol{S}_{i+1}^g|. \tag{5}$$

The storage space required ξ_1 should be

$$\xi_1 = m \times n + l \times l. \tag{6}$$

Since most of the space is shared with display memory, the stored space ξ_1 can be reduced to

$$\xi_1 = |\boldsymbol{S}_{i+1}^g| + |\boldsymbol{S}^r| - |\boldsymbol{S}_i^g \bigcap \boldsymbol{S}_{i+1}^g|. \tag{7}$$

The Inetcam does not require the above additional storage space, it however needs to transmit the maximum image data ρ_2, and

$$\rho_2 = |\boldsymbol{S}_i^g| + |\boldsymbol{S}_{i+1}^g| - |\boldsymbol{S}_i^g \bigcap \boldsymbol{S}_{i+1}^g|. \tag{8}$$

The number of pixels needed to update per frame η_2 is also

$$\eta_2 = |\boldsymbol{S}_i^g| + |\boldsymbol{S}_{i+1}^g| - |\boldsymbol{S}_i^g \bigcap \boldsymbol{S}_{i+1}^g|. \tag{9}$$

Example 4.1 Following the above example, and suppose 4-byte/pixel is used to represent the color information, then $\eta_1 = 98$ bytes, $\xi_1 = 98$ bytes, $\rho_1 = 3$ bytes, $\eta_2 = 98$ bytes, $\xi_2 = 0$ bytes and $\rho_2 = 392$ bytes.

From the above analysis, it is noted that the proposed algorithm can reduce the amount of data to be transmitted over the network at the cost of increasing the store space. The transmitted data amount over network plays an important in the image update performance. Is it also noted that the increased computation time and storage space amount are within acceptable limits.

Figure 4 shows the reconstructed image with the discussed algorithm. Figure 5 shows the retrieved image from Inetcam video server over the network. Inetcam uses video source with different resolutions and colors as shown in Table 1, and 5-frame per second is sent to the client.

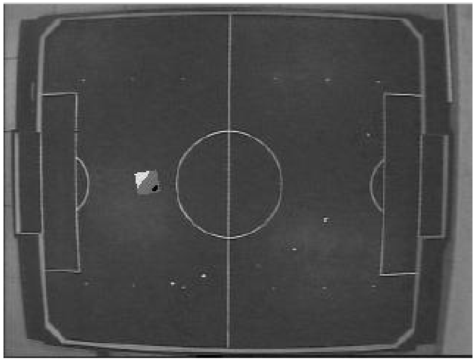

Figure 4: The reconstructed image using the proposed algorithm

From the stationary image comparison between Figure 4 and Figure 5, it shows that the proposed algorithm provides almost the real feel of the setup. In addition, within the NUS Intranet, the image update with the InetCam video server in Table 1 is compared. Since the proposed algorithm takes less than 150ms for the image update, the result shows that the algorithm has faster processing speed, and can provide a more practical interactive operation.

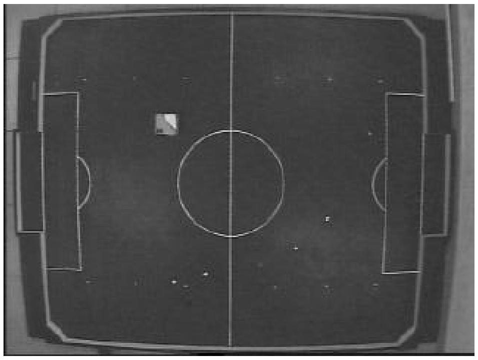

Figure 5: The received image from video server over network

Table 1: Average delay of InetCam video server

Format	*Resolution*	*Average Delay*
black-white	176 × 144	220 ms
QSIF(RGB24bit)	176 × 144	300 ms
SIF(RGB24bit)	352 × 288	740 ms
FULL(RGB24bit)	704 × 576	1250 ms

5 Conclusions

In this paper, a new image reconstructing algorithm over network for cyber robot motion is proposed. This approach provides a fast video update, especially over the narrow bandwidth network in order to transmit real time video feedback. Although the algorithm needs more computation time and storage space, less data transmission over network is required to reconstruct the image. Therefore this scheme provides the user with a faster video update and helps to improve the remote control performance over network. Through the web-based robot experiment, the algorithm has demonstrated the mentioned advantages.

References

[1] C. C. Ko, B. M. Chen, S. Y. Hu, V. Ramakrishnan, C. D. Cheng, Y. Zhuang and J. Chen, “A web-based virtual laboratory on a frequency modulation experiment”, *IEEE Transactions on Systems, Man, and Cybernetics,* in press.

[2] C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang and K. C. Tan, “ Development of a Web-based laboratory for control experiments on a coupled tank apparatus,” *IEEE Transactions on Education,* Vol. 44, No. 1, pp. 76-86, February 2001.

[3] C. C. Ko, B. M. Chen, S. H. Chen, V. Ramakrishnan, R. Chen, S. Y. Hu and Y. Zhuang, “ A large scale web-based virtual oscilloscope laboratory experiment,” *IEE Engineering Science and Education Journal,* Vol. 9, No. 2, pp. 69-76, April 2000.

[4] P. Vadakkepat, “Multi-agent systems: A survey from the robot soccer perspective,” *Int. J. Intelligent Automation and Soft Computing,* 6: (1) 3-17, 2000.

[5] P. Vadakkepat, “A hybrid control structure for vision based soccer robot system,” *Int. J. Intelligent Automation and Soft Computing,* 6: (1) 89-101 2000.

[6] P. Vadakkepat, “Vector field based path planning and Petri-net based role selection mechanism with Q-learning for soccer robots,” *Int. J. Intelligent Automation and Soft Computing,* 6: (1) 75-87 2000.

[7] G. Pegman, “Automation teleoperation,” *Industrial Robot*, vol.26, No.3, pp.184-187, Dec. 1999.

[8] C. Schmid, “A Remote Laboratory Using Virtual Reality on the Web,” *SIMULATION*, vol. 73, No.1, pp.13-21, Jul. 1999.

[9] W. J. Davis, X. Chen, A. Brook and F. A. Awad, “Implementing On-Line Simulation With the World Wide Web,” *SIMULATION*, vol.73, No.1, pp.40-54, Jul. 1999.

[10] J. Alex, B. Vikramaditya and B. J. Nelson, “Teleoperatoed Micromanipulation within a VRML Environment Using Java,” *Proc. of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems Victoria,* B. C., Canada, pp.1747-1752, 1998.

[11] J. Ballantyne and E. Wong, “A Virtual Environment Display for Teleoperated Excavation”, *Proc. of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems Victoria,* B. C., Canada, pp.1894-1899, 1998.