A Tele-robot Assistant for Remote Environment Management

Chunyuan Liao[1], Qiong Liu[2], Don Kimber[2], Surapong Lertsithichai [2]
[1]*Dept. of CS, Univ. of Maryland at College Park, U.S.A, liaomay@cs.umd.edu*
[2]*FX Palo Alto Laboratory, U.S.A., {liu, kimber, surapong}@fxpal.com*

Abstract

Using a machine to assist remote environment management can save people's time, effort, and traveling cost. This paper proposes a trainable mobile robot system, which allows people to watch a remote site through a set of cameras installed on the robot, drive the platform around, and control remote devices using mouse or pen based gestures performed in video windows. Furthermore, the robot can learn device operations when it is being used by humans. After being used for a while, the robot can automatically select device control interfaces, or launch a pre-defined operation sequence based on its sensory inputs.

1. Introduction

It is often desirable to manage a remote environment without being there. For example, a handicapped person may want to operate home appliances in multiple rooms when sitting on her/his bed; a scientist may try to explore a hostile environment remotely; a power plant engineer may prefer to manipulate distributed devices in a central station.

In this paper, we describe a remote environment management system based on a networked mobile robot. The robot is equipped with a set of cameras and microphones so that a user can drive it around to "see and hear". It can also electronically or mechanically operate on targets based on users' instructions. To simplify the management of remote targets, our system allows users to operate on remote targets seen in a live video window, with mouse or pen-based gestures performed in the video window. Furthermore, the robot can learn device operations from users. After being used for a while, the robot can automatically choose a control interface for a specific target ,or launch a pre-defined sequence of operations based on its sensory inputs.

Our system differs from conventional tele-conferencing systems [2][3][4][5][14][15]. in that it approaches remote device and object control through a video window served by a mobile camera. Unlike existing device control systems [6][7], our system does not require special devices for control, nor does it require people to be in the controlled environment, or the target to be networked. Moreover, our system does not need a human tele-actor, which was used in [11].

In contrast to systems in [1] and [10], video cameras in our system are installed on a mobile platform, giving us more flexibility to control devices at various locations. It raises challenging problems on dynamic device-action association. To the best of our knowledge, our system is the first one that uses interactive video, captured by a mobile platform, to manage a remote environment. Additionally, our system is trainable for remote environment management.

The rest of the paper is organized as follows. In the next 2 sections, we will present the system overview and the mechanism for remote environment management respectively. In section 4, we will discuss the trainable framework for automation. In section 5, feasibility experiments are reported. Finally we conclude our paper and discuss future directions.

2. System overview

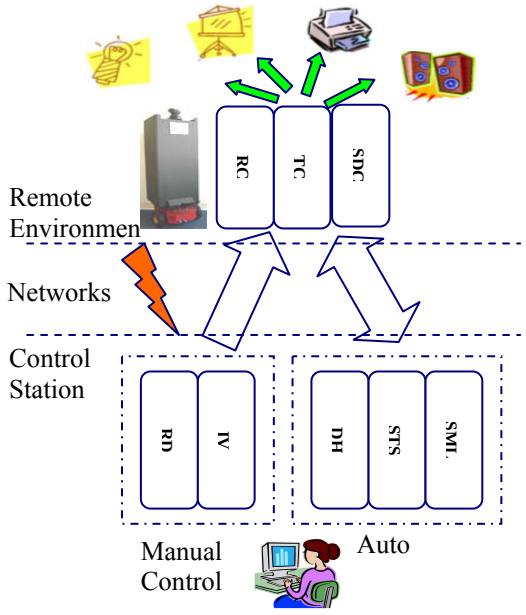

Figure 1. System Architecture

Figure 1 depicts the system architecture, which consists of a mobile robot platform and a control station. The robot and the control station are connected through Internet so that they can be physically far away from each other.

The functionalities of the robot platform can be divided into three modules: "Robot Control (**RC**)", "Sensor Data Collection (**SDC**)", and "Target Control (**TC**)". "**RC**"

receives commands from the control station and adjusts the robot's movement. "**SDC**" collects data from various sensors installed on the robot, and periodically sends the data back to the control station. "**TC**" manages targets electronically or physically on commands of users. For instance, if a mechanical arm is installed on the robot, "**TC**" can manage target movements by directly pushing or pulling the target, while a universal IR controller can be installed to easily control many existing home appliances. If the target is a networked device, "**TC**" may directly send commands to that device via network.

There are five modules in the control station. The "Robot Driving (**RD**)" and "Interactive Video (**IV**)" get user's inputs from a joystick, keyboard, mouse or pen and translate them into control commands for robot control or target manipulation. Meanwhile, the "Driving Helper (**DH**)" provides service on automatic collision avoidance and stall recovery based on received ultrasonic data. "Smart Target Select (**STS**)" and "Smart Macro Launch (**SML**)" can automatically launch specific controlling interfaces or macros based on system sensory inputs and the robot's past experiences.

3. Mechanism for Controlling remote targets through "Interactive Video"

3.1 Interactive Video

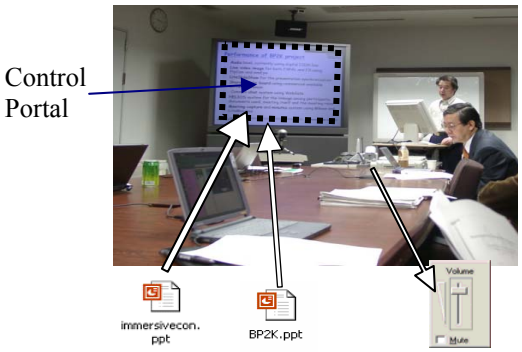

Figure 2. An illustration for Interactive Video

As mentioned before, our system supports "Interactive Video" [1]. This functionality is achieved by defining "control portals" in a live video window, and deploying corresponding control daemons on hosts. "Control portals" are active regions that can translate gestures performed in these regions to proper control commands. For example, the region occupied by the screen in figure 2 can be defined as a portal, and associated with a control daemon of the screen. With this definition, users' gestures performed in the portal can be translated into specific commands by the **IV** at control station, and forwarded to the remote daemon by the **TC** on the robot. With the screen portal defined for the video window, a person can drag a presentation file from her/his desktop to the portal and open that presentation on the remote screen. A user may also drag other people's presentation from the portal to his/her desktop [1] for note taking. With a similar portal defined for a microphone, a person should be able to adjust a remote microphone's volume for best sound quality.

3.2 Automation for Interactive Video in dynamic environment

Since cameras and devices are all fixed in [1], control portals in that system have fixed locations in the video window. This does not hold when the video window is served by cameras installed on a mobile platform.

To meet this challenge, our system provides the capability of defining "**dynamic portals**". With "dynamic portals", users only need to select a region containing a device seen in the current video window, and the "Smart Target Select" module will automatically identify the device based on sensor fusion, and associate it with the corresponding control daemon. Further, we can introduce "extra" gestures and/or action candidate list to compensate the imperfect auto-identification. In this way, users can manage remote devices in a similar way as they use fixed portals. In a device-rich environment, this automatic device identification process can quickly narrow the scope of possibly intended devices and greatly facilitate users.

Additionally, to help users manage a complex system, Automatic Interface Switch and Macro Launch are provided. After a user chooses the intended target, a specific control interface can be brought to the desktop of a control station. For example, when the robot approaches a screen, the control station can automatically launch a screen control panel for complicated screen manipulation. Meanwhile, users often have to repeat some operations on remote targets. To save people's efforts and improve efficiency, our system also allows a user to record an action sequence, called "Macro", and replay it later when the system sense similar inputs.

4. A Trainable framework for automation based on sensor fusion

4.1 Basic idea

Three kinds of automations are supported in our system. These automations are dynamic portal definition, target-specific interface switch, and automatic macro launch. The key is to establish associations between robot sensory data, which describes a target's surrounding circumstance, and automatic actions related to the target. This "association" can be achieved with a traditional classification approach, in which a sensory input vector can be labeled with a specific action id. In the training stage, the robot is driven to some places where a specific target can be properly seen and operated in the video window. Meanwhile, sensory data are collected and labeled with a proper action id. After obtaining enough training data for each action, the system will be able to initiate proper actions based on robot's sensory data.

4.2 Design of the Framework

The framework is based on sensor fusion techniques. Denote $\{A_i\}$ as the i-th action/action-sequence, $p(A_i \mid O)$ as the probability of taking action-Ai conditioned on environmental observation by m sensors $O = \{O_j, j = 1..m\}$. $p(A_i \mid O)$ may be estimated with

$$p(A_i \mid O) = \frac{p(O \mid A_i) \cdot p(A_i)}{p(O)}. \quad (1)$$

To minimize error, the action selection is based on

$$A_i = \arg\max_{\{A_i\}} \left(p(O \mid A_i) \cdot p(A_i) \right), \quad (2)$$

where $p(O \mid A_i)$ and $p(A_i)$ are estimated online based on users' past operations of the robot system. Assume various sensory data are conditional independent, the estimation of $p(O \mid A_i)$ can be achieved with

$$p(O \mid A_i) = \prod_j p(O_j \mid A_i), \quad (3)$$

where O_j is the sensory data captured by sensor j. Figure 3 shows the basic architecture of this framework. In this implementation, the correspondence of each sensory-reading/action pair is managed with a conditional probability learning and evaluation module. The prior knowledge of actions is also managed by various modules.

Currently, we only use the robot's location, heading and the color histogram of a user selected video region.

To further reduce wrong actions, the framework may also be requested to return a list of possible actions given the observation O, such that

$$p(O \mid A_{i_1}) \cdot p(A_{i_1}) \geq \Lambda \geq p(O \mid A_{i_k}) \cdot p(A_{i_k}) \geq \theta, \quad (4)$$

where θ is a threshold defined by users. Any action with scores below the threshold will be excluded from this list. With proper training, the candidate list can be short.

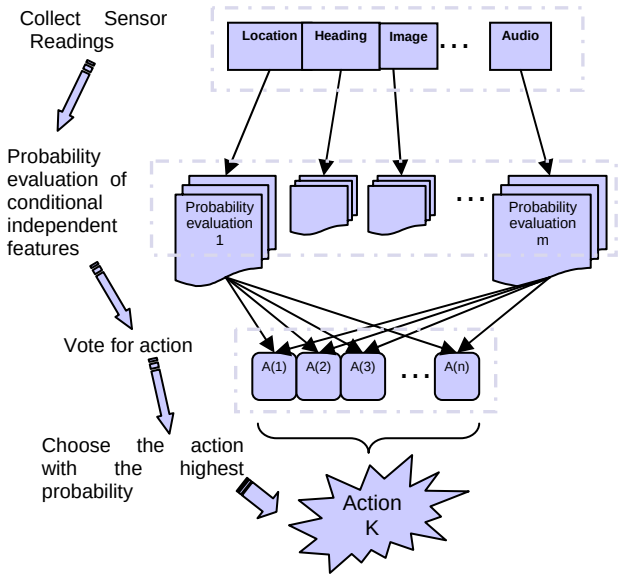

Figure 3. Trainable framework for automation

5. Feasibility Experiments

To verify the feasibility of our design, a series of experiments were conducted in our corporate conference room. Figure 4 illustrates the top view of the room layout, in which plasma display 1 (PD1), printer and podium, shown as three orange blocks, are chosen as the controllable targets for testing. The dot near the left of the figure is the origin of the top-view coordinates, and the up arrow is the robot's 0-degree heading direction.

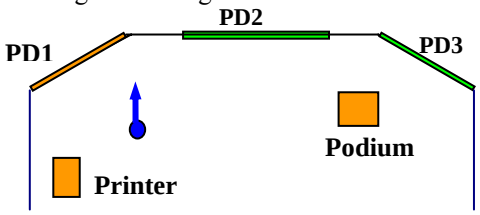

Figure 4. Experiment Room Layout

We trained the system with the robot location and direction data for these three targets. When a target appeared in our video window at a proper size, we performed an operation, for example, defining a portal or launching a macro, on the target. At that time, the system collects sensor data at frequency of 1 sample per second and labeled them with the action. On testing, when the robot approaches the targets and receives sensory inputs similar to those training data, the associated actions will be triggered. Thus it is basically a 3-class classification problem for the display, printer and podium respectively.

Based on the above setting, we examined the importance of each feature to classification and the accuracy of classification with given amount of training samples. The importance of feature f, or ratio of inter-class distance to the intra-class distance, is defined as

$$\text{imp}_f = \frac{(dist(m_1, m_2) + dist(m_2, m_3) + dist(m_1, m_3))/3}{(\sum_{c=1}^{3} (\sum_{i=1}^{n_c} dist(F_{ci}, m_c))/n_c)/3} \quad (5)$$

where m_i (i=1,2,3)is the centroid of class i, dist(x,y) is the Euclidian distance between point x and y in the space of feature f, n_c is the number of training samples of class c and F_{ci} is the i-th sample of training class c. In this experiment, there are only 3 classes of samples, corresponding to the display, printer and podium.

And the accuracy is defined as the ratio of number of correct classification results to the size of testing set. We use Monte Carlo methods for the accuracy estimation, in which we assume the testing samples satisfy Gaussian probability distribution. 100 trials are tested for each of the 3 classes.

The whole training set contains 543 samples. Beginning with 10% or 54 training samples, we expended the training set by 10% at each round. Figure 6 and 7 show the feature importance and classification accuracy with given training set size, in which the X axis represents the ratio of the actually used training samples to the total training set at

each step, while the Y axis is for feature importance or accuracy.

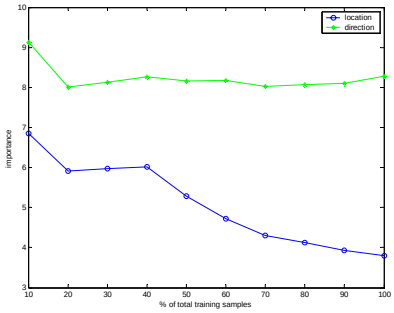

Figure6. Importance vs Training set size

The fig.6 shows that the feature "direction", the upper green line, remains high importance factor, more than 8, over all training sets, while the importance of "location", the lower blue line, goes down to less than 1. The reason is that training samples of classes gradually become more and more overlapped.

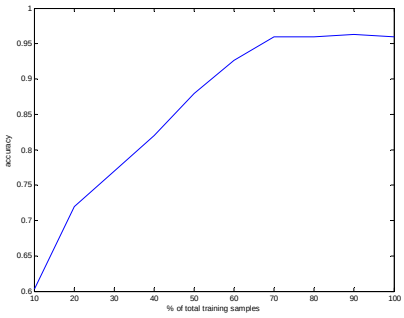

Figure 7. Accuracy vs Training set size

Figure 7 shows that the accuracy increases when the training set growing. With more than 50% of the total training sample, 272, we can get accuracy higher than 0.86.

The prototype worked very well on triggering proper actions in several talks and a poster exhibition. The results are positive and encouraging to our future research in this direction.

6. Conclusion and furture research

We have presented the design and implementation of a trainable mobile robot platform for remote environment management. Users can drive the robot to move around in a remote environment, and "see and hear" the scenes there. By using "Interactive Video", our system supports controlling remote devices by mouse/pen-based gestures performed in a live video window. Moreover, after the system is properly trained, it can help people define dynamic control portals. It can also automatically switch to a target-specific controlling interface, or launch pre-defined macros under certain conditions. In this way, users can save efforts in repeating the same actions.

Currently, the prototype system only supports electronic devices controllable via networks. In the future, we want to install a mechanical arm and a general IR controller on it to control more types of targets. Moreover, more image and acoustic features need to be included in the training for better performance.

Finally, it will be interesting to see a robot accumulate knowledge during its life-time.

Acknowledgements

The authors appreciate the generous support by FX Palo Alto Laboratory (FXPAL). Chunyuan Liao worked on this project during his summer internship at FXPAL.

References

1. Chunyuan Liao, Qiong Liu, Don Kimber, Patrick Chiu,Jonathan Foote, Lynn Wilcox. Shared Interactive Video, for Teleconferencing. *Proc. of ACM MM'03*, pp 546-554.
2. Microsoft Netmeeting home,

   http://www.microsoft.com/windows/netmeeting/
3. Web site of Polycom Corp. http://www.polycom.com/
4. Norman P. Jouppi, Wayne Mack, Subu Iyer etc. , First steps towards mutually-immersive mobile telepresence, *ACM CSCW 2002*, New Orleans, Louisiana, USA , p354 - 363
5. Web site for Prop. http://www.prop.org/
6. Khotake, N., Rekimoto, J., and Anzai, Y., InfoPoint: A Direct-Manipulation Device for Inter-Appliance Computing, http://www.csl.sony.co.jp/person/rekimoto/iac/
7. Web site of Makingthings Corp.

   http://www.makingthings.com/
8. VideoClix Authoring Software.

   http://www.videoclix.com/videoclix_main.html
9. iVast Studio SDK -- Author and Encode. http://www.ivast.com/products/studiosdk.html
10. Tani, M., Yamaashi, K., Tanikoshi K., Futakawa, M., and Tanifuji, S., OBJECT-ORIENTED VIDEO: INTERACTION WITH REAL-WORLD OBJECTS THROUGH LIVE VIDEO, *Proc. of ACM CHI92*, pp. 593-598, May 3-7, 1992.
11. Goldberg, K., Song, D.Z., and Levandowski, A. Collaborative Teleoperation Using Networked Spatial Dynamic Voting, *Proceedings of IEEE, Special issue on Networked Robots*, 91(3), pp. 430-439, March 2003.
12. Junyang, J. Weng and Y. Zhang, Developmental Robots: A New Paradigm, an invited paper in Proc. *Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Edinburgh, Scotland, August 10 - 11, 2002.
13. Web site of ActivMedia Corp. http://www.activmedia.com/
14. M. Chen, Design of a Virtual Auditorium, *Proc. of ACM Multimedia*, Ottawa, Canada, Sept. 2001, p19-p28
15. Web site of VRVS: "Virtual Room Video-Conferencing System",http://www.vrvs.org/About/index.html