Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER
UNCLASSIFIED	509683

TITLE

AUGMENTED REALITY TELE-MANIPULATION INTERFACE \(ARTEMIS\), DEMONSTRATION SITE SETUP FOR OPERATION THROUGH THE INTERNET

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to:

DCIEM No. 98-CR-44

Augmented Reality Tele-manipulation Interface (ARTEMIS), Demonstration Site Setup for Operation through the Internet

by

Shi Yin *
Julius Grodski +

* Image & Vision Systems
30 Godstone Rd., Suite 1202
Toronto, Ontario, Canada
M2J 3C6

Project Manager: Shi YIN, (416) 492-3318

PWGSC Contract No. W7711-6-7362/04-SRV

on behalf of
DEPARTMENT OF NATIONAL DEFENCE

as represented by

+ Defence and Civil Institute of Environmental Medicine
1133 Sheppard Avenue West
North York, Ontario, Canada
M3M 3B9

DCIEM Scientific Authority:
Julius Grodski
(416) 635-2085

Aug. 8, 1998

Contents

Executive Summary

The Augmented Reality Tele-Manipulation Interface System (ARTEMIS) is a technological capability which grew from its conceptual predecessor — the Augmented Reality through Graphic Overlays in Stereovideo (ARGOS™), developed under DCOEM sponsorship. ARTEMIS consists of a local operator site and a remote robot site. Both sites were implemented and located in the Ergonomics Teleoperation and Control Laboratory (ETC Lab) at the University of Toronto (UofT).

Under the current contract, Image & Vision Systems (IVS) undertook the task to develop a local operator ARTEMIS site on an SGI Indy Workstation and to modify the operation procedure. The purpose of this task was to establish a local operator site at DCIEM, thereby facilitating easier demonstration of ARTEMIS from any site external to UofT and for the one located at DCIEM. IVS completed the task. An operator site has been successfully established at DCIEM. The following major sub-tasks/modifications have been completed:

- Software porting to Indy Workstation
- Startup procedure modification
- Performance enhancements

Additional recommended enhancements to the system have been outlined.

1. Introduction

The Augmented Reality through Graphic Overlays in Stereovideo (ARGOS™)[1] was developed in the Ergonomics in Teleoperation and Control Laboratory (ETC Lab), at the University of Toronto (UofT), under DCIEM contract W7711-7-7009. The Augmented Reality Tele-Manipulation Interface System (ARTEMIS)[2] is a technological capability which grew from its conceptual predecessor — The ARGOS concept. ARTEMIS was developed at the ETC Laboratory in 1995 through the support from Manufacturing Research Corporation of Ontario (MRCO) and DCIEM. MRCO has been the holder of the Department o National Defence (DND) license for ARGOS.

A complete ARTEMIS system involves two sites, a local operator site and a remote robot site. The local operator site consists of input/display devices, the operator interface, and a computer workstation. The remote site consists of a robot arm, a stereo camera system and a computer workstation. Both sites are linked through Internet.

In the original design, both sites were implemented in the ETC Lab., at the University of Toronto. The local operator site involved an SGI Indigo2 workstation and the remote site involved a CRS bench top robot and an SGI Crimson workstation. Although tests have been carried out with temporary operator sites located as close as DCIEM and as far as Kyoto, Japan, the basic arrangement was confined to the specific configuration of the original design.

Under the current contract, W7711-6-7362/04-SRV, Image & Vision Systems (IVS) has undertaken the task to develop a local operator ARTEMIS site on an SGI Indy Workstation and to modify the operation procedure. The purpose of this task was to establish a local operator site at DCIEM, thereby facilitating easier demonstration of ARTEMIS from any site external to U of T and specifically from the one located at DCIEM.

IVS has identified and proposed three major sub-tasks in the contract proposal. They are: software porting from SGI Indigo^2 into SGI Indy, modification of the startup

procedure, and modifications to enhance system performance. IVS carried out the work and successfully completed the task.

2. Software Porting

The original ARTEMIS software was written in *C++* using *IRIS Inventor 1.0* (on the robot site), *Inventor 1.1.2* (on the operator site), *IRIS GL*, as well as *Motif*. The software was not compatible with the Indy OS IRIX 5.3. Special system software had to be installed to maximize the usage of existing code in the existing Indy Workstation by way of a minimum effort.

Open Inventor 2.0.1, irix4 Inventor 1.0.1, and *Inventor 1.1.2* have been alternatively installed and tested on the Indy workstation.

- Irix4 Inventor 1.0.1 should provide a common approach. However, since the Indy workstation is not under software support contract from Silicon Graphics, it would take too much effort to employ it.

- *Open Inventor 2.0.1* would provide the perfect solution to the issue, because it is the most current Inventor software. It is compatible with IRIX 5.3 and IRIX 6.x. If *Inventor 2.0.1* were used, the operator site could be directly used on a new O2 workstation. However, in order to use *Inventor 2.0.1,* a huge amount of work would have to be carried out to convert all the *IRIS GL* commands into *Open GL* commands.

- *Inventor 1.1.2* required least effort for implementation, and it was therefore chosen as a most pragmatic solution at this stage.

The ARTEMIS software employs double frame buffer. Since the existing Indy workstation was only equipped with an 8 bits graphics board, there were only 4 bits (16 colors) available for each buffer. Dithering technique was tested for displaying a full color (24 bits) image, however, appearance of the video image was not satisfactory. Therefore, a 24 bits graphics board was purchased and installed in the Indy Workstation to yield sufficient quality imagery.

The original ARTEMIS program used IRIS GL, which was an early version of the Open GL. Some *IRIS GL* commands are hardware dependent. These hardware dependent commands had to be identified, replaced by corresponding commands or by a group of commands which perform the same functions, or modified so that the program could perform correctly. Two such commands were ***pixmode*** and ***lrectzoom*** :

- Indy does not support ***pixmode***, which is used to switch from *wireframe* drawing to solid drawing. The command ***pixmode*** was replaced by an Inventor command ***SoDrawStyle***.

- ***lrectzoom*** is used to change the display size of an image. However, Indy does not support ***lrectzoom*** with a non-integer zoom factor. ***lrectzoom*** command was deleted and the display size was changed by another procedure.

3. *Startup procedure modification*

The original version of ARTEMIS was developed under the constrains in the development time. It demonstrated the principal features of the concept, but it did not encompass a user friendly interface. Consequently, anyone other than the original developer had difficulties in starting the program.

A new startup procedure had to be developed. The corresponding files, program structure, and environmental variables were modified. In order to simplify the procedure, a dedicated user account was established for ARTEMIS. By means of shell programming, the related startup steps have been grouped together as a batch process file. In this way, fewer steps are needed to start the program and to establish the connection between the Operator Site and the Robot Site in a semi-automatic process.

In the new startup procedure, certain information regarding the Robot Site is hard coded in the Operator Site. This information includes the host name and the IP address of the Robot Site computer; the user login name, password, and the file

structure of the Robot Site. In this way, the user does not need to know, or to remember all the tedious details. This simplifies the startup procedure.

It is to be noted that any changes in this information on the Robot Site need to be taken into account in the Operator Site. For details, refer to the instruction files.

Detailed instructions for the startup procedure are available on-line. They are stored in a number of files in the home directory. See the appendix for the copies of instruction files.

4. *Performance Improvement*

A number of modifications have been implemented in the source code level to improve the performance of the system. The three major enhancement areas were:

4.1 Graphic rendering speed increase

The SGI Indigo^2 employs more advanced graphics hardware than SGI Indy does. Therefore, the Indigo^2 rendering speed is much higher, especially to some commands, which are supported directly by the hardware as shown in the following. By means of the original rendering procedure of ARTEMIS program, it took 0.77 s for the Indy to execute a rendering loop (stereo wireframe robot and video image). This results in a rendering rate of 1.3 Hz. The rendering rate on the original Indigo^2 was about 10 Hz.

Two major modifications have been implemented with respect to Z-buffering and on-line image zoom. Z-buffer is not needed for wireframe graphics, therefore it has been disabled when wireframe robot is to be drawn. This modification reduces the render time by 0.13 s.

To display a frame of NTSC video on the full screen, a zoom action is needed. The original program performs such a zoom action on-line in every loop. The modified program calculates the zoom only once, immediately after the image is transferred from the Internet. Therefore, the zooming is not included in the rendering loop. This modification reduces the rendering time by 0.46 s.

In summary, after these modifications, it takes 0.18 s for a rendering loop, resulting in a rendering rate of 5.7 Hz.

4.2 Standalone version of ARTEMIS for the operator site

The original ARTEMIS has two demonstration formats. The full Internet ARTEMIS and the local ARTEMIS. The full Internet ARTEMIS is the one we mentioned above all the time, which consists of two sites communicating through the Internet. The local ARTEMIS runs only on the SGI Crimson workstation, which is connected with the robot. It was not possible to start the ARTEMIS only on the operator site. The program would crash.

IVS suggested that it would be advantages to develop a stand-alone version of ARTEMIS for the operator site. This program, residing in the computer at the DCIEM Operator Site, can be executed to demonstrate the ARTEMIS concept partially, without interfering with the Robot Site at the University of Toronto.

The standalone version of ARTEMIS has be successfully developed. All of the user interfaces in this standalone version are the same as they are the full version, except that one can not request a new video image or send a control command to the robot. The video images used in the standalone program are the images stored in the file from the last execution of the full version ARTEMIS.

4.3 Important bug fixing

In the original status of the software, when manipulating the graphical robot arm, there is a state where the wireframe model of the robot arm behaves in an unpredictable manner. It bounces back and forth, as in a dance. This "dancing" effect dramatically degrades the operational performance of the system.

Effort has been made to solve the "dancing" problem. A significant effort went toward setting up of proper constrains so that the singularity problem could be avoid. A bug in the file responsible for kinematics calculations has been fixed and thus the problem has been solved.

5. Conclusions

As required by the contract, IVS successfully developed the local operator ARTEMIS site on an SGI Indy Workstation located at DCIEM. Insight resulting from the work indicates that further work could be carried out in the following two respects:

- Currently, the NTSC analog video signal system, located at the Robot Site does not provide a reliable service. This problem needs to be solved when porting the Robot Site program from SGI Crimson to an SGI O2.

- Further modification on the user interface for program startup and manipulation need to be performed toward enhancement of the system to achieve higher performance levels regarding user's controllability.

References

[1] Paul Milgram, Anu Rastogi, Julius J. Grodski , "Telerobotic Control Using Augmented Reality", *Proceedings 4th IEEE International Workshop on Robot and Human Communication (RO-MAN'95) 5-7 July, 1995, Tokyo*

[2] Drascic, D., Grodski J.J., Milgram P., Ruffo K., Wong P., Zhai S., "ARGOS: A Display System for Augmenting Reality", *ACM SIGGRAPH Technical Video Review, Volume 88: InterCHI `93 Conf on Human Factors in Computing Systems,* Extended Abstract appears in Proceedings of INTERCHI'93: ACM Conference on Human Factors in Computing Systems, p 521, Amsterdam, April 1993

Appendix

On-line Instruction files

AboutARTEMI

README

HowToRunARTEMIS

HowToManipulateRobot

Instruction file AboutARTEMIS

```
##############################################
#                                            #
#        Welcome to ARTEMIS, Version 1.2     #
#                                            #
##############################################
```

There are two ARTEMIS demonstrations available, At the current directory, the home for the Operator Site of ARTEMIS. They are: local ARTEMIS and (full) ARTEMIS demonstrations.

(1) To run local ARTEMIS on the Operator Site, execute the program noSocket_ARTEMIS.

(2) To run (full) ARTEMIS, you need to setup ARTEMIS on the Robot Site of ARTEMIS in advance, and then execute the program connect_server. Refer to instruction file HowToRunARTEMIS for details.

When the program is started, the user interfaces are the same in both demonstration modes. To switch into the stereo display, press the key 'S' (Note: there are also other ways to do it too). To manipulate the robot arm, please refer to the instruction file HowToManipulateRobot.

If you can not remember the exact commands to manipulate the robot arm, e.g. to open the gripper and attach an object to the gripper, you might want to print the menu HowToManipulateRobot. Once the program is in robot manipulation mode, the whole screen of the computer is occupied, so you can not use on-line help anymore.

More information on the ARTEMIS concept and system can be found at http://gypsy.rose.utoronto.ca/people/anu_dir/papers/atc/atcDND.html

Instruction file README

```
##############################################
#                                            #
#        Welcome to ARTEMIS, Version 1.2     #
#                                            #
##############################################
```

This is the home directory for the Operator Site of ARTEMIS. There are following files in this directory:

AboutARTEMIS:
The general infomation regarding the ARTEMIS and how to run the demonstrations.

HowToManipulateRobot:
The instruction to manipulate the Robot arm.

HowToRunARTEMIS:
The instruction to run (full) ARTEMIS demonstration, where the robot arm in Universoty of Toronto is under control.

README: This file,

connect_server:
The program for connectting to the server, which is started on the Robot Site of ARTEMIS.

get_camera_data:
The program for retrieving camera calibration data from the Robot Site.

link_robot_site:
The program to establish the link to the Robot Site of ARTEMIS.

noSocket_ARTEMIS:
The program for running the local ARTEMIS demonstration on the Operator Site.

Instruction file HowToRunARTEMIS

```
##############################################
#                                            #
#        Welcome to ARTEMIS, Version 1.2     #
#                                            #
##############################################

    Instruction:  Running (full) ARTEMIS on Operator Site
```

When ARTEMIS has been properly setup on the robot site, including robot/camera homing/calibration, you are able to run (full) ARTEMIS on This Operator Site to remotely control the robot via Internet connection.
To setup ARTEMIS on robot site, please refer to the instruction file HowToSetupARTEMIS in the home directory of the Robot Site of ARTEMIS.

Operation procedure for running ARTEMIS:

(1) Connect the spaceball and stereo display device.

(2) If the cameras are re-calibrated, new camera data should be fetch over by executing a shell command:

```
[OperatorSite] % get_camera_data
```

(3) Connect to the remote Robot Site, by executing a shell command in the login directory:

```
[OperatorSite] % link_robot_site
```

A green window titled "Robot Site Terminal" will appear the upper-right Conner of the screen.

(4) Start the server at robot site, by executing a shell command in the green Robot Site Terminal:

```
[Robotsite] argos >> start_server
```

The following message will appear

```
The sock on the server was 4
    Listening for connect request on port 5123
```

(5) Connect to the server, by executing a shell command in a gray Operator Site Terminal:

```
[Operatorsite] % connect_server
```

Wait for the beep from the spaceball.

Note: Start_server program can only serve connect server once. So that when connect_server program stopped for any reason, start_server program has to be quit and restart before running next connect_server program.

(6) Click the right mouse button on the window with wireframe robot. Select the "Initialize Network" item from the Task Chunking menu.

The screen will be switched into stereo display mode with a robot frame superimposed onto a stereo video image.

The program has run into robot manipulation mode, by means of spaceball and keyboard.

Note: Most of the Robot Site information is hard coded in the Operator Site, and hence is hidden from the end user. Such information includes the host name and IP address on the Robot Site computer, the login name, password and file structure of the Robot Site. Any change of the above information need to be coped on the Operator Site. F

For detailed information about how such information is hard coded, please refer to the files: link_robot_site, get_camera_data, and .netrc

For further information about robot manipulation, please refer to the instruction HowToManipulateRobot.

To retrieve a new stereo image in JPEG format, click the right mouse button on the screen, which is in stereo display mode. Select the "JPEG video" item from the Video update menu. It could take long time to transfer image data from Robot Site to Operator Site.

Instruction file HowToManipulateRobot

```
##############################################
#                                            #
#        Welcome to ARTEMIS Version 1.2      #
#                                            #
##############################################
```

Instruction: How To Manipulate Robot -- Spaceball Control

Upon initialize, after the server is connected, the screen will switched into stereo mode, and there will be a beep sound. Once the black stereo video window with the robot frame coming out, press F9 with a beep from spaceball, then you can execute the spacebell control either on Robot site or on Operator Site. The various control functions through spaceball are as below:

Spaceball Control: (if do not be mentioned, it is not toggle switch)

E	: Virtual Robot end-effector, Default control mode
B	: The entire graphics moves for adjustment in registration
P	: Virtual Pointer
Y	: Virtual Object connected to the end-effector (Toggle switch)
Esc	: Quit the application

Robot Control: (if do not be mentioned, it is not toggle switch)

E	: Spaceball controls the virtual robot
F9	: Trajectory mode. ONLY mode in networked version
F8	: Flush trajectory points which are still at the local site (have not been sent to the robot)
F7	: Save a trajectory point. Maximum 8 points. There is a beep when the limit exceed and new point is not seen in the drawn trajectory
G	: Toggle the state of the gripper; open the jaws, close the jaws (Toggle switch)
F11	: send next trajectory point to the robot for execution
9	: Display Virtual Object attached to the end-effector (Toggle switch)
Y	: Control the virtual Object shape and transformation (Toggle switch)
Arrow keys	: Only in 'Y' mode to change the shape of virtual Object
F2	: gripper SPIN, virtual robot (Toggle switch)
F3	: display trajectory. Default ON (Toggle switch)
F4	: Gripper shape toggle between hollow two sided vs.single center plane (Toggle switch)

Virtual Object connected to the end-effector:

9 : Display Virtual Object attached to end-effector (Toggle switch)
Y : Control the shape and transformation of Virtual Object
E : control the end effector with spaceball again
Arrow key : change the shape of the Virtual Object (Cube).
These keys work only in the Y mode.

Screen Display:

S : Stereo mode. Hit 'S' again if there is some problem in switching (not toggle switch)
M : Mono mode(not toggle switch)
D : Dual mode(not toggle switch)

Video:

J : compressed Video(not toggle switch)
K : image differential compressed video(not toggle switch)

Graphics:

W : wireframe graphics (not toggle switch)
F : Fully rendered graphics (not toggle switch)
1-6 : Toggle display of link 1-6 (Toggle switch)
8 : Toggle the display of all the links from their previous state (Toggle switch)
N : do not show graphics on screen; only video (Toggle switch)
B : Move entire graphics to reduce registration error.
Move the spaceball while in this mode to move the robot and then immediately hit 'E' to put the spaceball back to robot control mode (not toggle switch)

Spaceball device:
Virtual Pointer (VP):

P : Set and display VP mode. VP display gets on as soon spaceball is touched (not toggle switch)
T : Take VP to new location and hit T. Take it new location and hit T again to display a new value at the bottom of the screen (not toggle switch)
Z : Hit Z to start creation of a virutal plane (no change on screen) Take virtual pointer to new locations and input these locations (3/4) with PICK button on the spaceball. After input of these 3/4 points with PICK button of spaceball is done. Hit Z again to tell the program that virtual plane has been created (sort of Toggle switch)
O : virtual encapsulator. Please do not use right now

Note:

* All key inputs are case insensitive. 's' and 'S' are same.
* If the key B is pressed and space ball is moved, it will displace the graphics.

The program can be re-registered by hitting key 'R'.

Dangerous Toggles:

G : gripper open and close
Y : virtual obejct manipulation

Network related details:

1. Compressed stereo JPEG file size: ~100Kbytes. JPEG compression time ~6 secs

2. Compressed stereo JPEG of difference in image:
~40 KBytes + 3 secs of extra processing for difference in image
3. Uncompressed stereo JPEG: ~1.8 Megs

5. Robot Packet size : 48 bytes

#509683