Performance Evaluation and Software Design for

EVA Robotic Assistant Stereo Vision Heads

Daniel DiPaolo
Trinity University
ER4
August 11, 2000

Kenneth Baker
ER4/Robotic Systems Technology Branch
Automation, Robotics, and Simulation Division
Engineering Directorate

Daniel D

Daniel DiPaolo

Baker

Ken Baker

Performance Evaluation and Software Design for EVA Robotic Assistant Stereo Vision Heads

Final Report

NASA/ASEE Summer Faculty Fellowship Program – 2000

Johnson Space Center

Prepared By:	Daniel DiPaolo
Academic Rank:	Undergraduate Student
University & Department:	Trinity University Dept. of Engineering Science San Antonio, Texas 78212

NASA / JSC

Directorate:	Engineering
Division:	Automation, Robotics and Simulation
Branch:	Robotic Systems Technology
JSC Colleague:	Kenneth Baker
Date Submitted:	August 11, 2000
Contract Number:	NAG 9-867

ABSTRACT

The purpose of this project was to aid the EVA Robotic Assistant project by evaluating and designing the necessary interfaces for two stereo vision heads – the TracLabs Biclops pan-tilt-verge head, and the Helpmate Zebra pan-tilt-verge head. The first half of the project consisted of designing the necessary software interface so that the other modules of the EVA Robotic Assistant had proper access to all of the functionalities offered by each of the stereovision heads. This half took most of the project time, due to a lack of ready-made CORBA drivers for either of the heads. Once this was overcome, the evaluation stage of the project began. The second half of the project was to take these interfaces and to evaluate each of the stereo vision heads in terms of usefulness to the project. In the key project areas such as stability and reliability, the Zebra pan-tilt-verge head came out on top. However, the Biclops did have many more advantages over the Zebra, such as: lower power consumption, faster communications, and a simpler, cleaner API. Overall, the Biclops pan-tilt-verge head outperformed the Zebra pan-tilt-verge head.

1 Introduction

The broad focus of the ERA[1] project is to provide an astronaut with a mobile assistant during EVA on another planet. The continuing example used for this project is one where the astronaut is a field geologist who is studying the surface of the planet, and this is what the project is focused on providing – a robot that can perform the duties necessary to assist a geologist on another planet. Very few assumptions are made about the planet by this project. Every other possibility has to be taken into consideration in the design of this robot. One of the more distinct and more likely possibilities is that the design produced by this project may someday produce an actual robotic assistant for an astronaut who must traverse the Martian surface. With this in mind, one must consider the rocky and unpredictable terrain of the Martian surface. Normally, this would suggest that a robot with a suspension system would behoove the project. However, before this was taken into consideration, the testbed robot was already purchased, and it in fact had *no suspension* at all. Also, it had been decided that a stereo vision system was going to be used as a key part in the guidance system of the robot. Of course, driving a robot with no suspension over rocky terrain does a great deal of damage to the viability of a stereo vision system because the gaze of the robot would constantly shift as a result of the rocky terrain.

That is where this project comes in. In conjunction with the stabilization and attitude prediction project of Dr. Kevin Nickels, the aim of this project was to help cancel out the negative effects of a non-smooth terrain on the performance of the stereo vision guidance system. To do so would require a high-performance pan-tilt-verge (PTV) head that could communicate with all of the other modules of the ERA project. However, the project already had two heads as candidates for use on the testbed robot – one was the Traclabs Biclops PTV head that was already being used as the development platform for our stereo tracking developer, Eric Huber, who had helped develop the head; the other was the Helpmate Zebra head which had been used in previous stereo tracking projects and, as such, already had some driver code written for it that we could use. However, neither of the heads had been field tested on the testbed robot while the stereo vision

[1] ERA = EVA Robotic Assistant

tracker was operating. So, before settling on one platform or another, an evaluation was necessary.

2 The Two PTV Heads

2.1 Traclabs' Biclops

The Traclabs Biclops PTV head was brought to the project by our stereo tracking developer, Eric Huber, who helped design the Biclops head before this project. Actually, the Biclops used for this project is one of the first functional prototypes for the design. The Biclops is a much smaller head than the Zebra, having dimensions of 155mm high by 160mm wide by 101mm long, and having a mass of 1.1kg without cameras attached. It uses a relatively standard cabling interface as well. The input into the Biclops PTV is a simple RS-232 serial port. However, within the Biclops head, there are two different voltages necessary for operation instead of the RS-232 standard of simply 7.5V and ground. The motors require a 24V power supply, and the RS-232 logic requires a 7.5V source. So, the serial connector takes input from a special power adapter which takes the normal serial connection from the computer *and* a 24V DC input. The other power requirement is that the Biclops may draw as much as 1A (ampere), with the motor pulling a maximum of 750mA and the logic circuitry pulling a constant 300mA.

The operation of the Biclops PTV head is fairly basic. The pan and tilt axis are driven by belts that are each controlled by an independent (not coupled) servo motor. The verge axis uses a metal spring which is drawn in and out by a motor-driven screw threaded through it. Each of these motors has an accompanying encoder that it uses to determine its position, velocity, and acceleration as well. The motors are independently controlled, with each having a PIC Servo Controller dedicated to it. The PIC controllers are in turn commanded by the Biclops firmware which was designed by Traclabs. Traclabs exposes a basic API (applications programming interface) for creation of drivers so that the Traclabs firmware in the Biclops can appropriately handle commands received via serial communication. So, once the software drivers are written, all one needs is a basic client program that takes advantages of these drivers and allows the user to command the Biclops head.

2.2 Helpmate's Zebra

The Helpmate Zebra PTV head had been used by the stereo image tracking team in previous projects, and because of the team's familiarity with the head and all of its capabilities (as well as the several Zebra heads available for this project), this PTV head was also selected as a candidate for the ERA project. This head is physically much larger than the Biclops head, and it is also shaped much differently. The Zebra PTV head consists of a tall base topped with a wide head. The base dimensions are: 3.875in. long by 2.25in. wide by 9.5in. high[2]. The head itself is 8.75in. wide, however, greatly adding to the dimensions of this PTV head. The communications and power scheme is similar to the Biclops. The Zebra communicates via a standard RS-232 serial port like the Biclops, and also like the Biclops, requires special power conversions to supply both logic power and motor power to the PTV head. Instead of a small inline connector adapter (like the Biclops), the Zebra instead has a controller box that, while greatly increasing the necessary space to operate the PTV head, affords a few other options as well. Under normal operation, the controller box requires two things to properly operate the PTV head – that motor power be enabled by the push of a button on startup, and that an emergency stop switch be plugged in. Both of these can be overridden, however, so as to automate the process much more. The controller box does very similar power conversions as well. It supplies the motors with 24V DC, and the logic circuitry with 5V DC (regulated). The current requirement is a bit steeper for the Zebra PTV head, 1.5A for the logic power and up to 6A for motor power. However, with this added power, the motors are given more torque, allowing faster movement of the head. The added power, however, has not proven either necessary or useful in our testing. For the majority of the functions required of the PTV head on this project (fixed gaze, slow motion tracking), the added velocity of the Zebra proves neither to be necessary or useful.

The Zebra head operates in a manner that is also very similar to the Biclops PTV head. The main difference between the two is that the Zebra has a pair of coupled axes – the pan and tilt axes. This means that any motion in the pan axis will cause a corresponding motion on the tilt axis. This has to be compensated for in the control of the PTV head. However, aside from that, the hardware operates essentially the same way

[2] *This is the total height, not just the base height*

as the Biclops does, with motors for each axis and encoders for each motor. Helpmate also gave a basic API for their Zebra PTV head for us to use, along with many little code snippets and examples. Utilizing those pieces of code, one can create the appropriate drivers to drive the head over the simple RS-232 serial port connection.

3 The Implementations

As mentioned in the abstract, the majority of time spent on this project was on the design and programming of the software for each of the two PTV heads. The ERA project requires that many other software modules of the project be able to communicate with the PTV head as well as one another, and to do such things, it was decided that using CORBA (Common Object Request Broker Architecture, a communications protocol) on the two Linux computers mounted on the robot was to be the way of interfacing all of the components together. So, it was necessary to design drivers for the heads that would be able to receive CORBA calls, interpret them appropriately and then issue the necessary calls to the PTV head in the head's own language via Linux serial port commands. Also, one of the requests of the project is that there be a standard PTV CORBA API so that everyone would be able to communicate with *either* of the PTV heads in the same fashion. So, to any CORBA clients, it is transparent as to which PTV head you are actually communicating.

3.1 CORBA Basics

CORBA is the proverbial "glue" that unifies all of the modules of this project into a cohesive unit. Using CORBA and its services, the ERA project can successfully run a number of different servers that interact with one another simultaneously and even on different computers running different operating systems. CORBA is simply another layer of abstraction that sits on top of other layers that handle things that are normally important to servers: initialization, requests, exceptions, etc. Any CORBA interface is well-defined by the IDL (Interface Definition Language) file of that interface. So, to establish the transparency described above, it was necessary to establish a standard set of commands that could be done by both PTV heads and to create a standard IDL.

However, since the other end of the server (that had to communicate with the PTV via serial commands) was *necessarily* different between the two heads due to the different APIs of each, the layers underneath the CORBA IDL had to be customized.

3.2 The Biclops Implementation

The Biclops PTV head was developed under a Windows environment, and with a Windows environment in mind. As such, the only code available for the Biclops head was not Linux compliant (due to uses of MFC (Microsoft Foundation Classes) functions among other things). However, the Windows code was two-layered in nature, using two separate C++ classes to describe a Biclops object. The first class was a simple *Serial* class that provided nothing more than a few simple functionalities for serial connections such as: opening/closing a port, setting port parameters, sending/receiving data, etc. This class was mostly unusable due to the heavy use of MFC function calls. The second class was a more general *SerialBiclops* class which contained an instance of the *Serial* class, as well as general calls to actually drive the motion of the PTV head. This code was relatively portable, as many of the calls were either using the *Serial* pointer class member or doing other simple operations. What needed to be done to get this working was to essentially use existing serial code from the Zebra PTV head (whose development began before the Biclops's development), and to turn it into a reasonable facsimile of the original *Serial* class that was implemented in the Windows NT code.

3.3 The Zebra Implementation

The Zebra head was not designed with a particular computing platform in mind. The documentation for the Zebra provides fairly portable (though very basic) code as a jumping off point for the coding of a driver for the PTV head. However, the portability of the code often encumbered it with many things that were unnecessary to the project. For example, the code provided by Helpmate is mostly written in C instead of C++. It is definitely more portable in that respect but it is not nearly as easy to work with as C++ which is object oriented. It also makes using that code to create a CORBA client even more difficult because the interface system of CORBA is made much simpler when dealing with objects defined by classes instead of dealing with a collection of functions.

However, the code for the Zebra was still divided in a very similar fashion. Most of the low-level serial code was written to be independent of the code that actually runs the head itself. It would not be a great undertaking to rewrite the code into C++ form, but there was not enough time to do so.

3.4 Comparison

Although the Biclops was actually developed with a different operating system in mind, implementing its functionalities into a Linux CORBA server was actually significantly simpler than it was for the Zebra. The usage of C++ in the code provided by Traclabs greatly helped the development of the Biclops driver. By separating the serial code and the Biclops-specific code, it was easy to overcome the Windows-specific serial code and to plug in working Linux serial functions. With the Zebra, although the code was already designed to work on Linux, the code that was provided was all very low-level code with no higher-level interface, and therefore required much more design for that driver than the Biclops did, as Traclabs provided code for some of the simpler and more useful functions like moving.

4 Performance Evaluation

Once the drivers for each of the heads was completed and ready to be used on the ERA robot, the actual performance evaluation began. Keeping in mind that the purpose of these PTV heads is to be used for the stereo tracking software that is currently being developed for this project, certain things are of greater importance in terms of performance. The main traits that were tested were:

1. Response time
2. Smoothness of move

4.1 Response Time

Response time is always an important issue in performance evaluations, but even more so with this project because of the nature of stereo tracking. Stereo tracking depends upon the ability to know where the object of interest is through correlations made by looking at the video provided by the cameras mounted on the PTV head. The tracker then sends commands telling the head where to look next in order to try to center the object of interest in the video (that is, in the PTV's gaze). However, without an adequate response time from the PTV head, the object of interest may go out of frame. Once the object of interest is out of frame, it is impossible to track it, and the only way tracking can continue is through an exhaustive search of the area in an attempt to reacquire the object.

To gauge head response time, we will record the amount of time between the issuing of a move command and the time visible movement is seen. The easiest and most reliable way to record the latter is to have a utility analyzing the video stream which will record the time at which it shifts significantly enough. However, since that utility is not available at the time of this writing, we had to rely upon a keypress from someone watching the head.

The setup for the tests was:

- PTV head – mounted on the robot, connected to serial port on robot computer
- PTV server – running on robot computer
- PTV client – running remotely, across a wireless network connection, on a PC

As it turns out, even with a wireless network connection setup, the heads respond much more quickly than a person can react and press a key. The fastest response that was measured was approximately 200ms for each head. So, 200ms was essentially the resolution of our test, but the response time of each head was better than our resolution could measure.

4.2 Smoothness of move

For the purpose of this project, a smooth move is a very big plus. With smoother moves, tracking is much easier and much more efficient. This is because with the way the stereo tracking is done is that it likes to begin searching for the object of interest at or near the last known position of the object. The more accurate the PTV head is at moving to the goal and not overshooting it, the more likely it is that tracking will be successful.

The test designed to measure the smoothness of each head's moves was that each head was to be set to move at a fairly high speed, and that it would make a significant move at that speed while we log the position of the head and then analyze it – computing overshoot and settling time for each, and then comparing them.

The setup for these tests was identical to the response time tests:

- PTV head – mounted on the robot, connected to serial port on robot computer
- PTV server – running on robot computer
- PTV client – running remotely, across a wireless network connection, on a PC

As mentioned in the description of the test, this test was supposed to use a high-speed move. However, the Zebra server was not to the point where the velocities were easily modified, so we had to use the default velocities, giving a slower move, and hence, giving it an advantage in this test because it is less likely to overshoot at a slower speed.

For this test, only the pan and tilt axes were measured for overshoot for two reasons. First, we aren't using vergence, and second, because the movements in vergence are generally so small that any overshoot is almost trivial. The move performed was a simultaneous 45 degree move in both the pan and tilt axes, with the Biclops moving at its maximum velocity on both axes and the Zebra moving at its default velocities (due to the problems mentioned above).

The results of the test confirmed the hypothesis for the Zebra head. There was no measurable overshoot in its movement at the sampling rate we used – the maximum for that head, about 100Hz (this value was obtained through experimentation). The Biclops, of course, did have some overshoot. It overshot on the pan axis by 0.6073° and on the tilt axis by 0.4183°, settling to the eventual goal in 1.24s and 2.02s, respectively. These results are fairly good. The overshoot in the pan axis only translates to 0.032m (3.2cm) shift at a verge depth of 3m, the amount we are currently using, and the tilt overshoot only translates to 0.022m (2.2cm) shift at that verge depth. Within those settling times, the tracking software can more than make up for that small a shift.

So, the Zebra's results don't really say much, since how the move was relatively slow. The Biclops results, however, confirm that it performs relatively well, with an acceptable amount of overshoot.

5 Conclusions

The first objective of the program, to design a common software interface for the two stereo vision heads, was very successful. It has proven to be fairly reliable in the constant testing it has undergone, and has proven to be a complete enough system for use in the upcoming field tests in Arizona.

The second objective of the program, the evaluation of each of the stereo vision heads using the common software interface, was quite telling. The performance evaluation tests of each head didn't reveal much, as both performed fairly equally. The Zebra head's best quality was its superior reliability, with its track record of zero failures sharply contrasting the three Biclops failures encountered during the design and testing. However, the other evaluations such as considerations of power, size, and API showed the Biclops to be a clear winner. Its lower power consumption (without reducing the responsiveness to a substandard level), more compact size, and much more amenable API put the Biclops ahead of the Zebra in just about every other category. As a result, the Biclops has been chosen by the ERA project team as the head to be used for the field tests in Arizona.