Development of a Stereoscopic Robot Platform Utilizing an Education in Engineering and Computer Science

Cameron Carper[1] [Chih-Hao Wu[2]]

Abstract – A research project, titled "Development of Machine Vision Algorithms for Cooperation in Multi-Agent Robotic Systems," was proposed to investigate methods of developing a robot platform capable of cooperating with other robotic entities using stereoscopic cameras. Though it is complex to implement vision algorithms for robotics, such a solution is highly flexible. Once in place, the systems for machine vision can easily be adapted to almost any environment: land, water, or even extraterrestrial landscapes. Another greatly desired effect of this system is that of autonomy. By allowing a machine to visually analyze its environment and react accordingly, human interaction can be eliminated. This is of paramount importance in hostile environments or any area where human presence is unlikely. This autonomy also has the ability to greatly reduce the amount of data transmission to and from the robot by allowing the robot to make decisions through its visual logic software instead of relying on a human controller.

Keywords: Robotics, machine vision, stereoscopic

PURPOSE

The purpose of this paper is to act as a parallel to undergraduate research, performed by the author, aimed at developing a robot platform capable of cooperating with other entities of a multi-agent robotic system using stereoscopic vision. The project outlines design decisions and their rationale. This approach allows a reader not only to follow the design process of the author, but also provides educational demonstration of the principles that make the design possible. The project is also designed to be entirely modular, providing libraries and reconfigurable systems which allow future researchers to easily incorporate existing work into new projects.

HARDWARE

Platform

The platform, shown in Figure 1 below, for the robot is designed specifically for simplicity, strength, and hardware accommodation. Aluminum is used as the primary construction material due to its availability, workability, low cost, and strength. The chassis components were created by a local machinist and arc welded together. The chassis measures 58.5 cm in length, 48.5 cm in width, and 47.5 cm at the tallest point. The frame is made entirely of tubular aluminum with the top and bottom enclosed by 0.6 cm thick aluminum sheets. A shelf has been constructed within the body to accommodate a mid-sized Personal Computer (PC) case required for logic calculations (Figure 1, Point A). An 11.5 cm square is outlined with the same tubular aluminum used in the frame and covered with an aluminum sheet to which motors can be mounted (Figure 1, Point B). Before this assembly, the sheets were drilled to provide necessary mounts and shaft throughput for the motors. These mounting locations are located at the bottom rear of the chassis (Figure 1, Point B). For simplicity, a single swivel caster is centered on the front of the platform such that the rear-mounted motors may provide direction control (Figure 1, Point C).

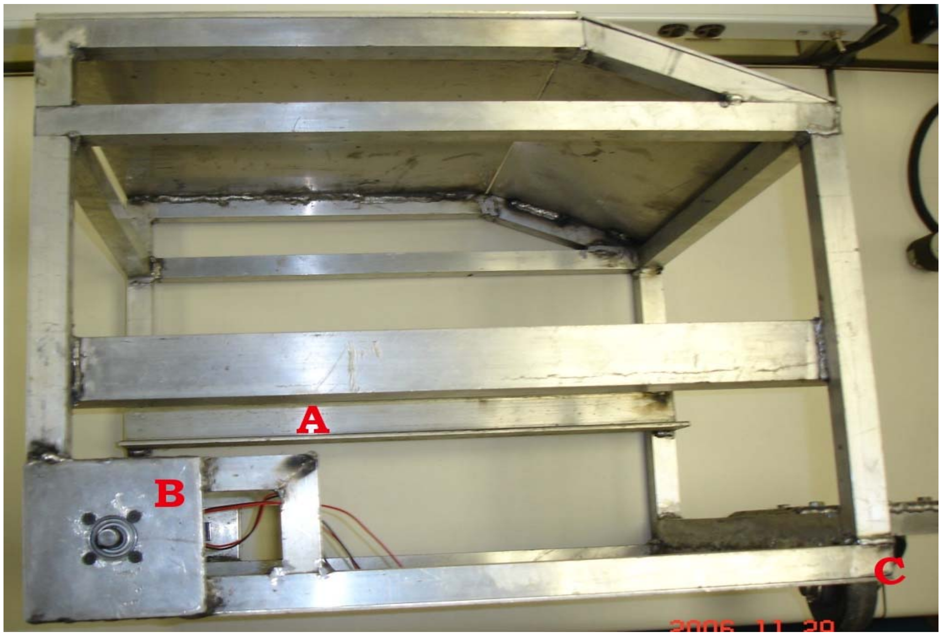

Figure 1 - Robot Platform

Stereoscopic Cameras

The cameras used to obtain sensory input are a pair of Point Grey Research Dragonfly digital video cameras. Each camera operates at 640x480 pixel resolution with 16 bit color depth at 30 frames per second. The video stream is transmitted as Bayer encoded grayscale images [6]. Data communication with the PC is accomplished through an IEEE1394 (Firewire) interface. Each camera generates approximately 16.5 Megabytes of data per second – well within the 400 Megabit (50 Megabyte) per second bandwidth of IEEE1394. Because each camera in the stereoscopic system provides a separate video feed, an IEEE1394 controller card is utilized in the PC to provide each camera with an exclusive data bus.

Motors

Basic locomotion is accomplished by two Dunkermotoren GR-63 DC brush-type motors fitted with RE30-2-500 two channel incremental encoders and PLG52 planetary gears. The motors operate at 12 volts and the attached encoder measures speed and direction at 500 ticks per revolution [2]. This encoder configuration allows data to be reported at every tenth of a centimeter for the current wheel configuration. The planetary gears provide 20.25:1 reduction which is ample for the current application. The aforementioned configuration was chosen for adherence to the proposed platform as well as the generality which allows reuse of components should the hardware configuration change.

Motion Controller

The motion controller, which interprets commands from the PC and directs the motors to take appropriate action, is a JRKERR PIC-SERVO SC 3PH capable of driving 6 Ampere (A) continuously at 12 – 48 volts DC [3]. This higher power board was chosen to match the 5.2 A requirement of the Dunkermotoren motors. To accomplish appropriate heat dissipation for the National Semiconductor LMD18200 integrated circuits, used for amplification, a custom aluminum heat sink was fabricated. The need for active cooling, via a fan in the project case, is being investigated. The use of two channel encoders allows the controller to calculate velocity and distance traveled which is instrumental in determining the robots position. PC to controller communication is accomplished through a full duplex RS-485 network [3]. The host PC transmits instructions via a dedicated transmit wire pair and receives data through a shared pair of receive wires. While the controller circuitry could have been constructed manually from discrete components, purchasing a prefabricated solution was feasible due to funds available and the complexity of the circuit. Courses such as Introduction to Engineering and Electric Circuits provide the education necessary to correctly configure the motion controllers.

Project Case

The two controller boards were placed in a customized project case for protection and ease of transportation as seen below in Figure 2. The project case measures 20.32 x 15.24 x 7.62 cm and has been custom fitted with power and data inputs and outputs. Screw down power terminals were used for power input and output to allow for simple, but sturdy, connection and disconnection from the platform. The RS-485 input, for PC data communication, is supplied via a DB-9 port which is the connector specified by the RS-485 standard. Data input for the incremental encoders was customized to a DB-9 connector due to availability, simplicity, and standardization. With all inputs and outputs provided removable connectors, the controller boards can easily be removed from the platform for maintenance or storage. Once the controller unit is functioning correctly it can be abstracted from the development process and regarded as a completed module. It can therefore be viewed as one self-contained unit which demonstrates the value of abstraction and modularity in systems design.

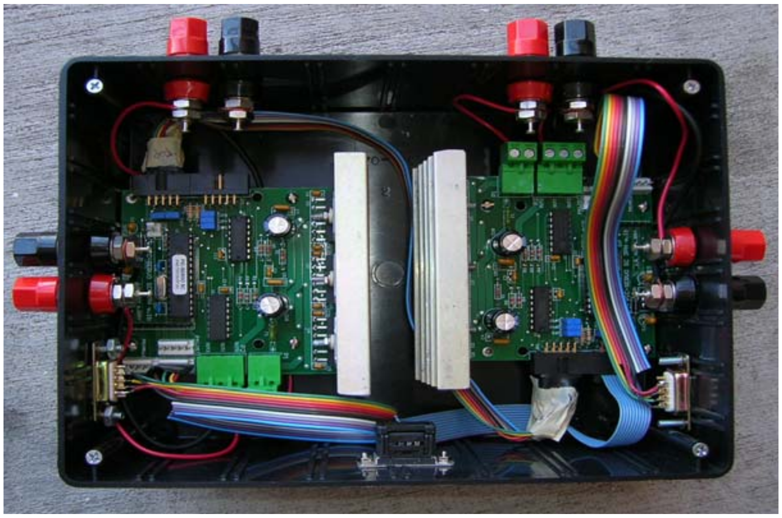

Figure 2 – Motion Controllers in Case

Computer Hardware

The current computing system consists of an x86 architecture PC utilizing a micro-ATX motherboard with an AMD Athlon64 3500+ 2.2 GHz processor. 512MB of DDR SDRAM memory is currently available, but may be expanded if needed. A PCI IEEE1394 controller card has been added to accommodate the stereoscopic cameras. Also, an Axxon MAP/950 PCI RS-422/485 serial card has been added in order to allow communication with the motion controllers without the need for a RS-232 to RS-485 converter. A 60GB hard drive is used to ease development but will be removed upon software completion to be replaced with bootable, removable media.

SOFTWARE

Operating System

Gentoo, a Linux distribution, was chosen as the operating system for the PC due to its availability, modularity, and cost. Gentoo, like most Linux distributions, can be used free of charge under the GNU General Public License. The modularity of Linux allows users to tailor the system to their exact specifications. A Graphical User Interface (GUI) is currently being used for testing, but will be eliminated as the information processing programs can run on a terminal. The removal of the GUI will allow more of the computing resources to be dedicated to image and logic processing. The availability of the operating system source code, as well as abundant information on the Internet, provides a wide base upon which to build the interfaces required for information gathering and processing. The selection of Linux as the proper operating system for the project was based on knowledge from an Operating Systems course and personal experience of the author.

Controller Interface Module

In order to maintain modularity, a controller interface module is used to abstract hardware specific code from the trajectory plotting module. Besides the immediate affect of producing more understandable programming structures, this abstraction allows the trajectory plotting software to work with other motion control hardware seamlessly. Instead of reimplementation of all project code, a new controller interface can developed to replace the JRKERR specific interface. Obviously, the new interface must maintain the same methods as the one it replaces in order to allow for transparent substitution.

JRKERR provides Networked Modular Control (NMC) libraries and program examples for Microsoft Windows, but not for Linux. However, Brian Rudy has developed libnmc which is a port of the NMC library that supports the Linux operating system [7]. libnmc is written in C, a common language for engineers and computer scientists. This familiarity allows the user to comfortably make use of the library and alter it should the need arise. Communication with the motion controllers is achieved via serial data transfer over the RS-485 network.

Camera Interface Module

In order to provide a simple software interface to the cameras through the IEEE1394 connection, the libdc1394 library is utilized. libdc1394 constructs another module, and therefore another layer of abstraction, which ensures that the robotic software core is flexible and easily portable. The low-level hardware interaction is performed by the library delivering a polished object from raw data. The libdc1394 library also applies the Bayer conversion in order interpret the grayscale, Bayer encoded image as a color image [1]. libdc1394, like libnmc, is written in C due to its close relation to system hardware and integral need for quick computations.

Image Processing Module

Image processing is also placed into a separate module for abstraction, an obvious requirement due to its incredible complexity. Gandalf, another open source library, is used for basic image processing. Gandalf was developed specifically as a computer vision and numerical algorithm library and is therefore written in C because of the speed required of such calculation-intensive computing [5]. After the images are passed through Gandalf, a custom developed C-based program, attempts to detect a specific target or marker within the image. Once the marker is found in the image from one camera, it is compared with an image from the second camera in order to determine the distance and direction of the object. The current image processing routines scan the image for a specific color, unique in the environment, and specific shape that will provide the most accurate information for position determination and trajectory plotting. The marker used most effectively in testing is a green square.

The image processing module analyzes a single frame from one of the cameras to find the average concentration of green colorization of each pixel by iterating through every pixel and storing the green value. Next, a standard arithmetic mean is applied to the collected color data to find the average green concentration. Then, the frame is iterated through, pixel by pixel, to determine which areas have a higher than average green value. A color tolerance can be used in the calculations if the targeted color is not unique and found elsewhere, to some degree, within the frame.

The addresses of the pixels with higher than average concentration of green colorization are set in a basic binary image. This binary image is then filtered, removing latent noise, and leaving an image containing only the robot's target. These advanced algorithms and the data structures required to facilitate their use are made possible by Computer Science classes such as Data Structures and Advanced Data Structures.

Logic Module

Using the images produced by the image processing module, the angle to the target can be determined by calculating the distance between the center of the camera image and the center of target. Next, the distortion of the target marker is measured to calculate the aspect angle to the target. These measurements are used to determine the proper path towards the target. In order to dock successfully, the distance must also be calculated. This is achieved by applying the same process to the corresponding image from the second camera. That is, the image that was taken at the same time as the one previously processed. Once both images have been processed, stereovision formulas are applied to derive the distance to the target. The angle and distance are then combined to determine the location of the target.

The calculated location is then used to determine the robot's path to the target. A simple algorithm, shown below in pseudo-code, has been developed and is currently being tested [4].

```
while (target is too far away) {
      if (aspect angle within limits) {
            if (target is not centered) {
                  turn robot towards target;
            } else {
                  move robot forward;
      } else {   //Aspect angle too high
            move robot backward;
              }
}
dock with target;
```

Figure 3 – Trajectory Determination Pseudo-code

CONCLUSION

Data Flow

As seen through the description of the software, each module occupies an integral role in solving the larger problem by contributing useful information to the process. The camera interface takes raw data from the cameras and translates it into data that can be readily manipulated by the image processing module. The image processing module filters all the extraneous data from the frame and delivers a simplified image to the logic processing module. The logic processing module then calculates the distance and trajectory to the target and issues a high-level command to the motion controller interface. This high-level command is translated into the proper low-level commands or electrical signals which will direct the motor to take the action dictated by the logic processing module. The data flow between these modules can be seen in Figure 4. The overall organization of the project was accomplished by project planning and management skills learned in Computer Science classes such as Systems Analysis and Design I and Systems Analysis and Design II.

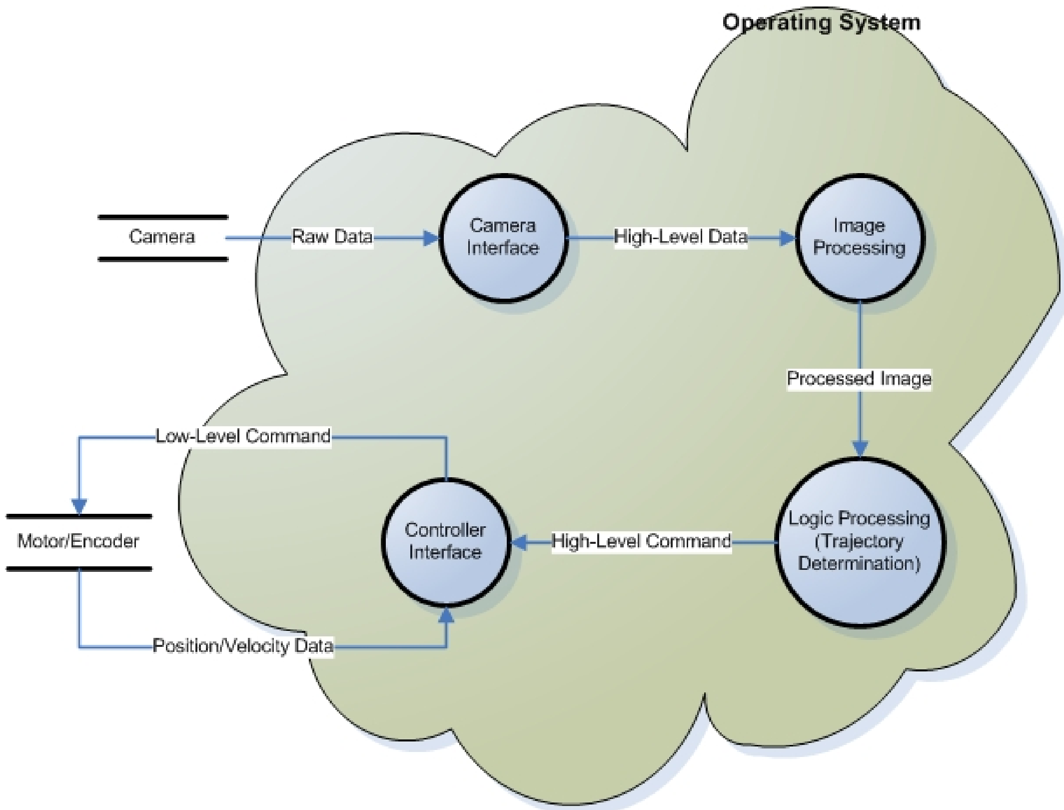

Figure 4 – Conceptual Data Flow

Progress

The hardware platform for this project has recently been completed. However, thorough testing of the motion controller module has yet to be conducted. The image processing module will be optimized to provide more flexible target detection and more efficient use of computing resources. Logic software is in continued development with expectation to be completed by January, 2007. At that time, a functional prototype will be constructed for practical testing and improvement. The project is planned, and on schedule, to be completed in May, 2007.

REFERENCES

[1] Douxchamps, Damien. *libdc1394: The Linux API for IEEE 1394 / Firewire Cameras*. Retrieved January 2005, from http://libdc1394.sourceforge.net

[2] Dunkermotoren. *Dunkermotoren – Brushed DC Motors GR/G*. Retrieved January, 2005, from http://www.dunkermotoren.de/default.asp?id=9&mid=11&lang=2

[3] Kerr, Jeffrey. *PIC-SERVO SC 3PH Motion Control Board*. Retrieved January, 2005, from http://jrkerr.com/pssc3ph.pdf

[4] Liddell, Kyle and Chih-Hao Wu, "Development of an Open Source Software Package for Autonomous Robotic Docking Using Stereoscopic Imaging," *Proceedings of the 2006 Midwest Section Conference of the American Society for Engineering Education*, 2006, pg 4.

[5] McLauchlan, Philip. *Gandalf: The Fast Computer Vision and Numerical Library*. Retrieved January, 2005, from http://gandalf-library.sourceforge.net

[6] Point Grey Research. *IEEE-1394 digital camera*. Retrieved January, 2005, from http://www.ptgrey.com/products/dragonfly/dragonfly.pdf

[7] Rudy, Brian. *Blue Cube*. Retrieved January, 2005, from http://www.praecogito.com/~brudy/blue_cube/code/blue_move/

Cameron Carper

A senior at Arkansas Tech University, Cameron will graduate in May, 2007 with a B.S. in Computer Science. His immediate plans upon graduation involve an overseas internship. On his return to the United States, Cameron plans to pursue a M.S. in Computer Science and a B.S. in Electrical Engineering.

Chih-Hao Wu

Dr. Wu is currently served as Assistant Professor of Electrical Engineering Department at Arkansas Tech University. Dr. Wu graduated from University of Texas at Arlington of Arlington, Texas at 1995, 1996, and 2000 with Bachelor of Science, Master of Science and Ph.D. all in Electrical Engineering.

---

[1] Arkansas Tech University, 1815 Coliseum Drive Russellville, AR 72801, darc@darchorizons.com

[2] Arkansas Tech University, 1815 Coliseum Drive Russellville, AR 72801, cwu@atu.edu