An Integrated Mobile Robot System For Testing Vision Algorithms

Extended Summary

by

H. H. Yakali,[1,2] D. Raviv [1,2] and M. Herman [2]

[1]Electrical Engineering Department and The Robotics Center,
Florida Atlantic University, Boca Raton, FL 33431

[2]Robot Systems Division,
National Institute of Standards and Technology
Gaithersburg, MD 20899

1. Introduction

We describe an integrated real-time vision-based closed-loop system which will serve as a platform for testing vision-based navigation algorithms. The system consists of several subsystems: a real-time image processor and a camera, a mobile robot and a workstation. We detail each of the subsystems, and the communications between them. Finally, we describe an algorithm used to test the system and discuss future work.

Figure 1 is a diagram of the overall system. A camera on top of the robot is connected to the video input stage of the image processor which digitizes and processes the images and outputs the results to the workstation for high level image processing. The output of the workstation is a set of motion commands to the robot.

2. System Components

2.1 Image Processor and Camera

The image processor we use is the Pipelined Image Processing Engine, PIPE [1], which is capable of processing standard size images (256x256 pixels) in one frame time (1/60 of a second). PIPE has an input stage, modular processing stages (MPS), and an output stage (Fig.2). A PIPE with 8 modular processing stages is capable of executing 1.2 billion arithmetic and 7.36 billion Boolean sum-of-products operations per second. PIPE has special processing units with built-in operations like addition, subtraction, AND, OR, XOR, single-value functions (sin, square root, etc.), and two-value functions (atan(a/b), etc.). A PC-AT is used as a host computer to program PIPE.

Identification of commercial equipment in this paper is only for adequate description of our work. It does not imply recommendation by the National Institute of Standards and Technology, nor that this equipment was necessarily the best available for the purpose.

Third Conference on Recent Advances in Robotics, Florida Atlantic University, Boca Raton, FL, July 2-3, 1990.

There are four major ways of processing images for feature extraction in PIPE:

- As isolated points: *Point processing,*
- As spatial neighborhoods: *Spatial Processing,*
- As a sequence of images: *Sequence Processing,*
- As a stack of eight Boolean bit plane images: *Boolean Processing.*

These four types of processing operators can be used individually or in any combination. The following quote is taken from the PIPE Manual[1]:

"*What does PIPE do?*

PIPE is a multi-stage parallel processor designed to do one thing very well: the processing of images at video rates. It is not a general purpose parallel processor. Each stage in the system is designed so that all input, processing and output are completely synchronous with the video raster and are done in one TV field time (1/60th of a second). This processing is performed with the goal of extracting features from the input image stream. Some typical features include:

- *Edges and gradients in gray scale or chromaticity,*
- *Intensity changes over time and/or space,*
- *Motion and other dynamic events,*
- *Stereo or range maps,*
- *Angular orientation of edges or gradients,*
- *Corners or spots,*
- *Isolated points.*"

Further details about PIPE may be found in [1,2].

The CCD video camera mounted on the robot is connected to one of the video input stages of PIPE. We work with 256x256 images digitized into 256 gray levels (8-bits).

2.2 Mobile Robot

The DENNING Mobile Robot [3] is composed of three major subsystems: mechanical, hardware, and software (Fig.3).

The mechanical subsystem consists of a frame, wheels, motors, and batteries. The robot is powered by three 12V rechargeable batteries. It has three wheels and two motors, one for steering and the other for driving. A head plate on the top of the main frame turns together with the wheels while the main frame does not rotate even when a turn command is executed by the robot.

There are four major components in the hardware subsystem (Fig.4): computers, ultrasonic sensors, motor control system, and electric power system. The main computer is a 68008 board. This board communicates with the user through an RS-232 serial port. A second serial port permits direct connection to a host computer, such as a VAX or SUN. The ultrasonic sensors and the motors are controlled by a dedicated Z-80 board. Available voltage sources on the robot are +5V,±15V, and +36V. The robot is controlled by translation and rotation commands. There are two modes for specifying a motion command. One of the modes, involving *move* and *turn* commands, uses distance, speed, and acceleration. The second mode involves *drive* and *steer* commands and uses speed, and acceleration only. Translation commands are specified in units of 1/10 of a foot, and rotation commands in units of 1/10 of a degree. The maximum allowable distance, velocity, and acceleration for a single command are 25 ft, 4.4 ft/s, and 4 ft/s^2, respectively. The maximum allowable angular displacement, angular speed, and angular acceleration are 360°, 57°/s, and 172°/s^2, respectively.

The software subsystem interprets data from the RS-232 serial port and issues necessary commands to other modules, such as the navigation and the sensor/motor module (Fig.4).

2.3 Workstation

The SUN uses an 68020 CPU and operates under the UNIX Operating System. Since this operating system runs several system tasks, the CPU is never dedicated fully to any user program. To avoid this problem we plan to use a dedicated 68020 board that will be installed in the SUN.

Although the PIPE performs many image processing algorithms very efficiently, there are many high level image processing algorithms that PIPE cannot run. These algorithms are run on the SUN 3/160 workstation. The SUN also runs planning and control algorithms for the DENNING.

3. Integration of the Subsystems

In this section we describe communication between the PIPE and the SUN, communication between the SUN and the DENNING, and integration of the PIPE, SUN, and DENNING.

3.1 Communication Between PIPE and SUN

Our communication program is written in the C language and runs on the SUN. This program uses *haltpipe*, *startpipe*, and *steppipe* commands to control the PIPE. Data transfer between the PIPE and the SUN is done through the VME Interface Bus. In order to achieve reliable data transfer, synchronization between the PIPE and the SUN has to be established. This is done by reading the data while PIPE is in halt mode. After this reading process is done, the PIPE sequencer is stepped using the *steppipe* command, that is, PIPE executes the next instruction and then halts.

3.2 Communication Between SUN and DENNING

Communication between the SUN and the DENNING occurs over an RS-232 serial port. This communication program opens a 9600-baud communication port for the DENNING. In order to test the communication, a C program is written to control the DENNING from the keyboard. In this program motion control commands entered by the user through the keyboard are sent to DENNING to control the speed and the direction. The user can change the speed and direction of motion in increments or decrements of 0.2ft/s and 10°/s, respectively.

3.3 Integration of PIPE, SUN and DENNING

In our current integrated system, the camera is mounted on the head plate (Fig.3). From the control point of view, we have a digital control system with two sampling rates. These are T_1, the rate at which images are sampled by the PIPE, and T_2, the rate at which commands are transmitted to the DENNING (Fig.5).

The value of T_1 includes digitizing and reading 4 center lines from an image and processing the image. The former takes 17.6 ms. The latter takes a variable amount of time depending on the image processing algorithm. We have chosen to make the value of T_2 greater than or equal to 300 ms since this seems to result in the smoother motion for the DENNING. Since the execution of motion commands in the DENNING involves mechanical components, it is significantly slower than the PIPE. If there is enough time to process another image before the decision is sent to the DENNING, a command is transmitted to PIPE to send a new image to the SUN. The overall closed-loop system can be analyzed using multi-rate digital control theory; we intend to do this in the future.

Figure 6 shows the general flowchart of the control algorithm running on the SUN. After per-forming initialization, this algorithm goes through two kinds of loops. The inner loop involves sending an image read command to the PIPE, running an image processing algorithm, determining whether a decision about the next command to the DENNING should be made, and repeating these steps if a decision is not made. The outer loop includes the inner loop, but sends the command decision to the DENNING before reading the next image from the PIPE. T_1 and T_2 are controlled by the SUN and only T_1 is a function of the decision algorithm. In that sense the SUN acts as a digital controller for the system, and these loops can be considered to be control loops.

These two loops can be controlled independently. Since the value of T_1 is much smaller than T_2, the inner loop can be executed many times before the decision algorithm is run. This permits the decision to be based on many frames of image data.

4. Procedures, Algorithms and Real-Time Considerations

A simple vision-based tracking algorithm has been implemented to test the system. The task is to track a dark vertical line moving either to the left or to the right on a white background. The camera tracks the line by rotating the head plate. Figure 7 shows the test set up and the vision algorithm. The values Δ_P, Δ_S, and Δ_D represent delays caused by processing on the PIPE, SUN, and DENNING, respectively.

As shown in Fig.6, other decision algorithms can be plugged into the loop. A constraint for the decision criterion is to force T_2 not to be less than 300 ms.

In the current implementation, only the four central scan lines of each image are read from the PIPE. This takes 17.6 ms. Then the gray level values of all four pixels in each column are summed and scaled, forming a new line image. The algorithm then searches for the maximum change in brightness in this image. The position of this change is an edge point. A turn command is sent to the DENNING so as to rotate the camera on the head plate towards this edge point. The angular velocity for the turn command is determined as a function of the distance of the edge from the center of the image. The value of the angular velocity is zero, if the distance is between -10 and 10 pixels; ±2°/s, if the distance between ±10 and ±50 pixels; ±6°/s, if the distance greater than ±50 pixels.

5. Conclusions

This paper has presented a vision-based closed-loop system which has been integrated and tested. This system will serve as a test bed for evaluating navigation and mobility algorithms such as obstacle avoidance algorithms. A test algorithm for tracking of a dark object on a white background has also been described.

We plan to develop navigation algorithms that can handle both stationary obstacles or targets and moving ones, and which do not use a-priori knowledge about the environment. There are several approaches that can be used, such as optical flow[4,5], stereo image analysis[6], laser range scanners[7], time-based approach[8], qualitative approach[9], camera fixation[10,11], neural networks[12], etc. We plan to concentrate on the time-based and qualitative approaches as well as on the camera fixation approach.

References

1 .. The PIPE User's Manual, Aspex Incorporated, 1987. 530 Broadway, New York, NY 10012.

2 .. Kent, E.W., Shneier, M.O., and Lumia, R., "PIPE (Pipelined Image Processing Engine)," J. Parallel and Distributed Computing, 1985.

3 .. "DRV-1 Product Manual, Main Text: Setup, Operation, & Description," Denning Mobile Robotics, Inc., 21 Cummings Park, Woburn, Massachusetts 01801.

4 .. Horn, B.K.P. and Schunk, B.G., "Determining Optical Flow," Artificial Intelligence, vol. 17,1981.

5 .. Rangachar, R., Hong, T.-H., Herman, M., Luck, R.L. and Lupo, J., "Real-Time Differential Range Estimation Based on Time-Space Imagery Using PIPE," Proc. SPIE Real-Time Image Processing II: Algorithms, Architectures, and Applications, Orlando, Florida, April 1990.

6 .. Abbott, A.L. and Ahuja, N. "Surface Reconstruction by Dynamic Integration of Focus, Camera Vergence, and Stereo," IEEE Second International Conference on Computer Vision, Tampa, Florida, 1988.

7 .. Jarvis,R.A., "A perspective on the Range Finding Techniques For Computer Vision," IEEE Transactions PAMI, vol. 5, March 1983.

8 .. Raviv, D., "Invariants in Visual Motion," The Robotics Center and The Electrical Engineering Department, Florida Atlantic University, 1990.

9 .. Nelson, R.C. and Aloimonos, J., "Using Flow Field Divergence For Obstacle Avoidance Towards Qualitative Vision," IEEE Second International Conference on Computer Vision, Tampa, Florida, 1988.

10 .. Raviv, D., Herman, M. "Towards an Understanding of Camera Fixation," IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, May 1990.

11 .. Ballard, D.H. and Ozcandarli A., "Eye Fixation and Early Vision: Kinematic Depth," IEEE Second International Conference on Computer Vision, Tampa, Florida, 1988.

12 .. Zhou, Y.T. and Chellappa, "Stereo Matching Using a Neural Network," In Proc. Intl. Conf. on Acoustic, Speech, and Signal Processing, New York, NY, April 1988.

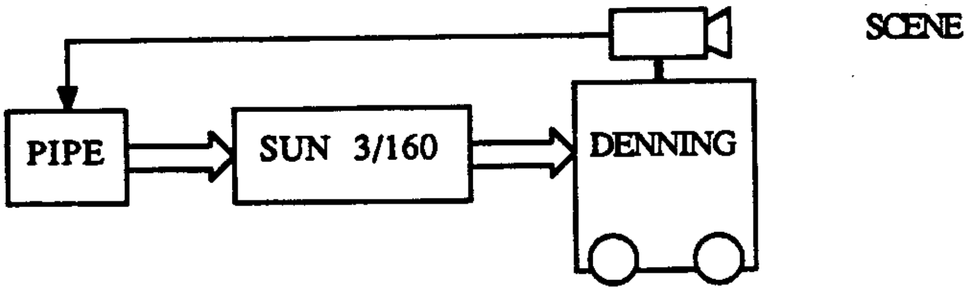

Fig.1 The overall system.

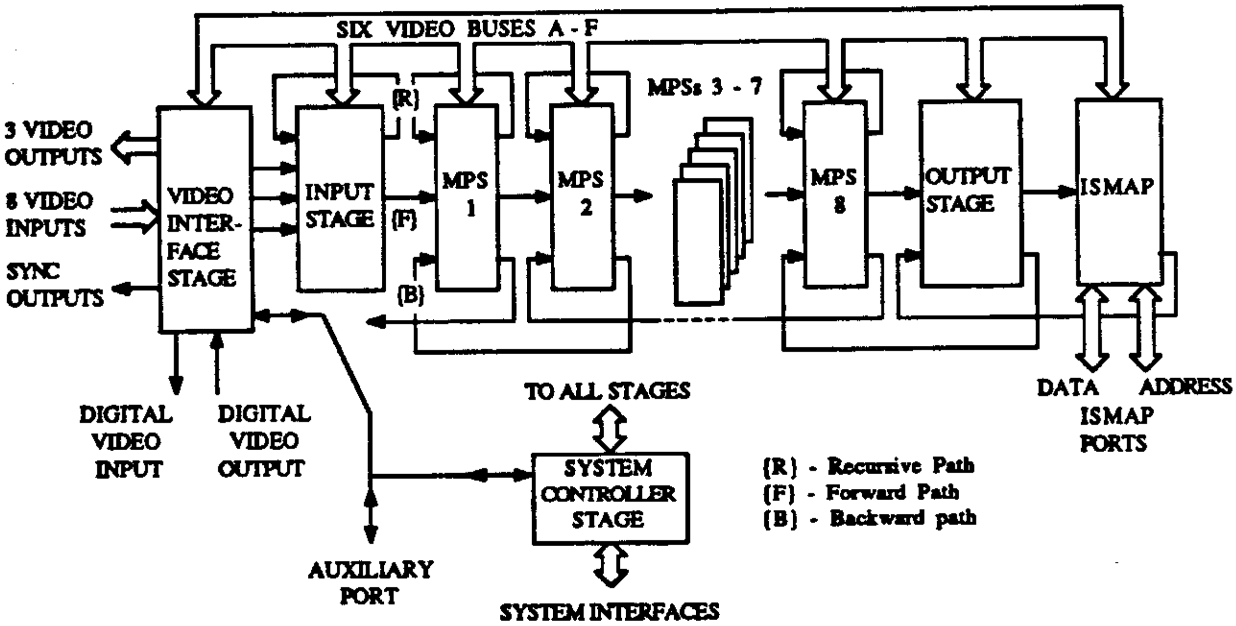

Fig.2 PIPE Hardware description.

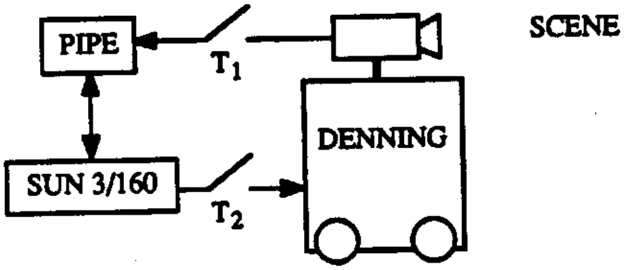

Fig.5 Sampling times shown in the closed-loop system.

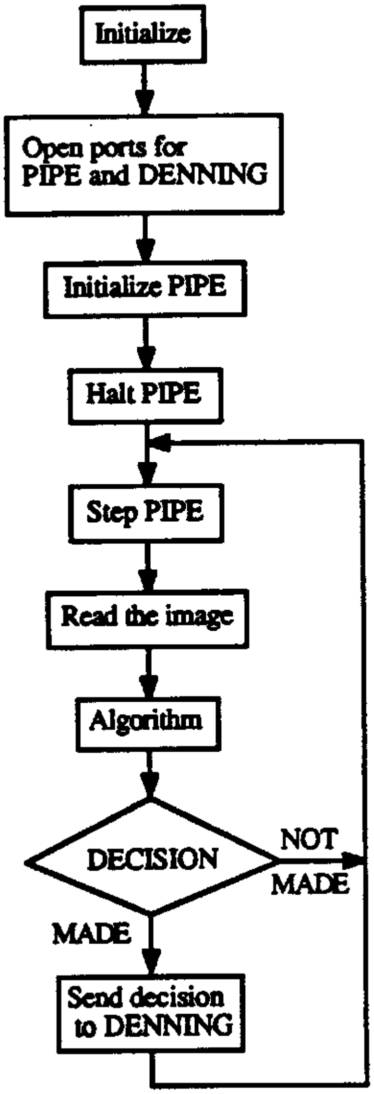

Fig.6 Flowchart of the software.

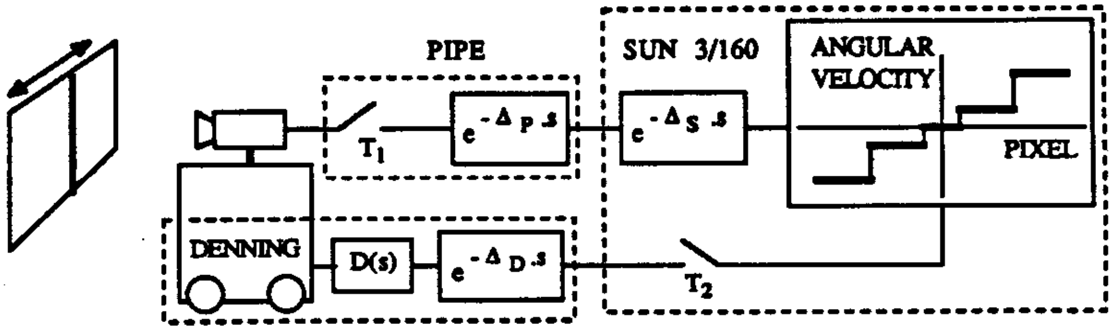

Fig.7 Block diagram of the closed-loop system.

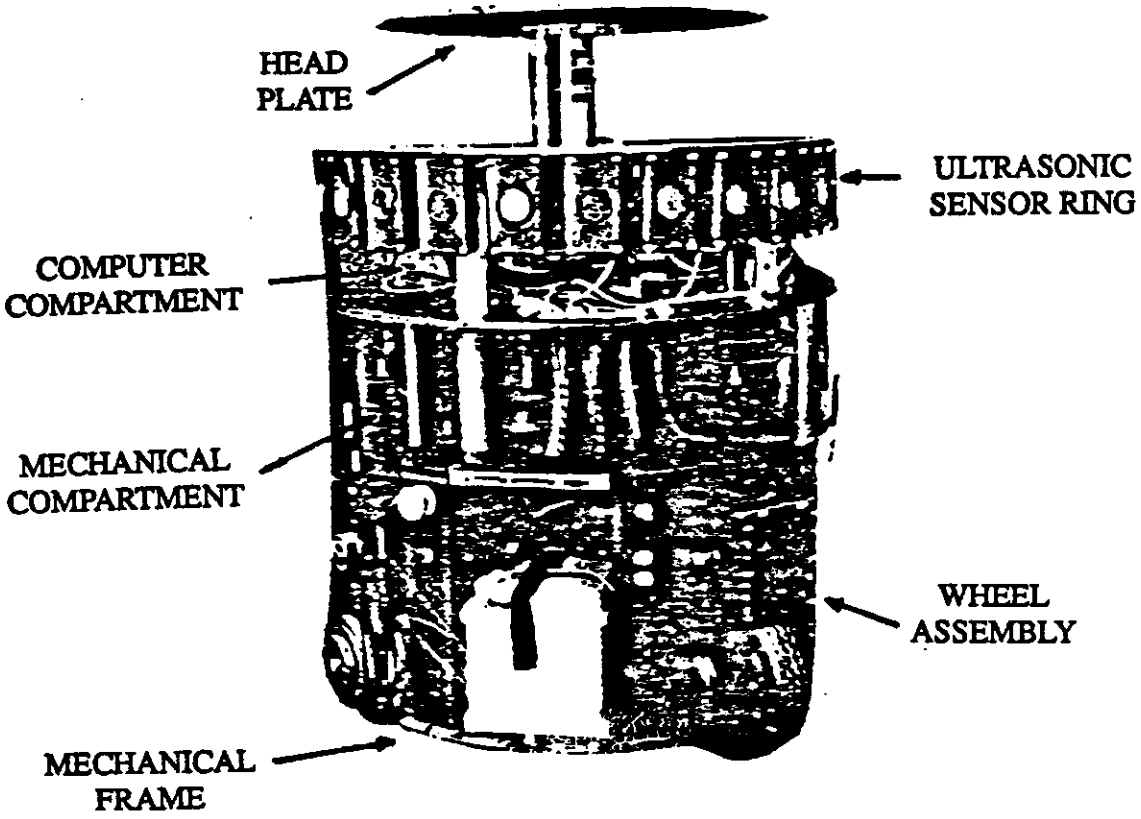

Fig.3 Denning Mobile Robot.

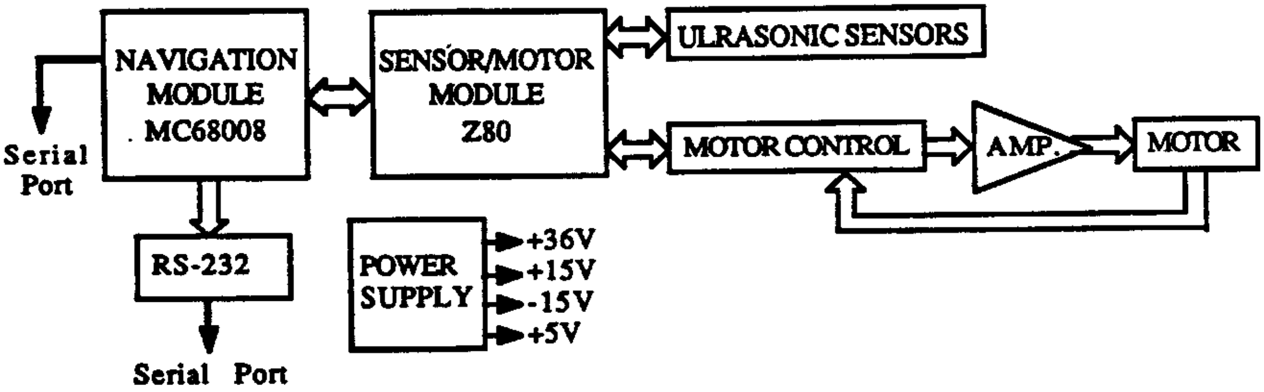

Fig.4 Hardware components of DENNING mobile robot.