A Low Cost, Online, Computer Controlled Robot Architecture Using a CCTV Network as Sensors

Akhil Piplani, Abhinav Gupta and Shailendra Mahani
Department of Electronics and Communication, Jaypee Institute of Information Technology University, Noida, India

***Abstract*—An important deterrent in deployment of robots in factories across the globe is the cost involved. A rather large part of that cost is the sensors which are also very unreliable and least robust. This paper proposes a cheaper, simpler and more dynamic robot prototype by using the factory's existing CCTV cameras as the primary sensing equipment. Image Processing and total control of the robot(s) is handled by a computer. Using a server-client model, the robot(s) can also be controlled online. The software for the prototype was developed in MATLAB. This system is easily reprogrammable and deployable in a vast range of situations.**

***Index Terms*—cameras, robot programming, robot sensing systems, image processing, online control.**

I. INTRODUCTION

Modern day factories across the globe employ millions of robots. However, for many products, it is still cheaper to employ humans instead of robots. Clearly, the major deterrent in deployment of robots is the cost.

A rather large part of that cost is the sensors involved. Sensors still remain the least durable and also the least reliable component of our robots. A major bottleneck for the progress in robotics is that many pilot projects in the field of sensors and sensor feedback have been discontinued. However, without appropriate sensors, a robot remains dumb and inflexible[1].

Even with all the progress in robotics, the progress is relatively slow. A key reason for the slow growth is the lack of standardization. Robotics companies have no standard development approach or software to control the robots with. Each project starts at square one.

Clearly there is much scope for reduction in development, production and deployment costs. This paper proposes a simple and generalized approach at developing robot while using a more cheaper and durable set of sensors, the CCTV cameras. Most factories already have a CCTV network installed for monitoring, security and safety reasons. This paper proposes to use this existing system as the primary sensors for the robots. The only additional cost is that of the computer that controls all the robots. An additional feature gained by using this is the ability to centrally control and reprogram all the robots. As a result the cost spent on several sensors and control hardware located on the robots is saved. It is rooted on the fact that the cost of processing power in a computer is much cheaper than that of the sensors and control hardware located on the robot.

II. THE PROTOTYPE SYSTEM

This project was aimed at development of a cheaper, simpler and more dynamic robot prototype by using a CCTV camera as the primary sensing equipment. Image Processing and total control of the robot is handled by a computer. This system is easily reprogrammable and deployable in a vast range of situations.

The prototype developed is primarily a mobile robot that avoids obstacles in its path. This robot employs stepper motors for their accuracy without the need for a closed loop control system. The computer obtains an image of the robot's surroundings. The computer divides the whole image into cells and categorizes them as filled and empty. Finally, the robot is moved across a safe path through the empty cell to its destination.

The prototype developed has the following characteristics:

Camera: CCTV

Camera Interface: USB

Programming language: MATLAB

Robot: Differential Drive, Stepper Motors

Robot Interface: Parallel Port

This system significantly reduces costs by utilizing the factory's existing CCTV network. Other advantages include central control of several robots and the resulting easy re-programmability.

III. THE ROBOT

The robot we made is a mobile robot that uses differential drive. It is best to keep the design of mobile robots as symmetric as possible[2]. This ensures that the centre of mass remains at the physical centre. Apart from symmetrical design, we also designed the robot to keep the centre of mass as low as possible. It uses two stepper motors as the actuators. Two ball castors are present for support.

The robot uses stepper motors as actuators for their accuracy without the need for a closed loop system which are more expensive.

Fig 1, 2, 3 represent the blueprints of the robot that was designed.

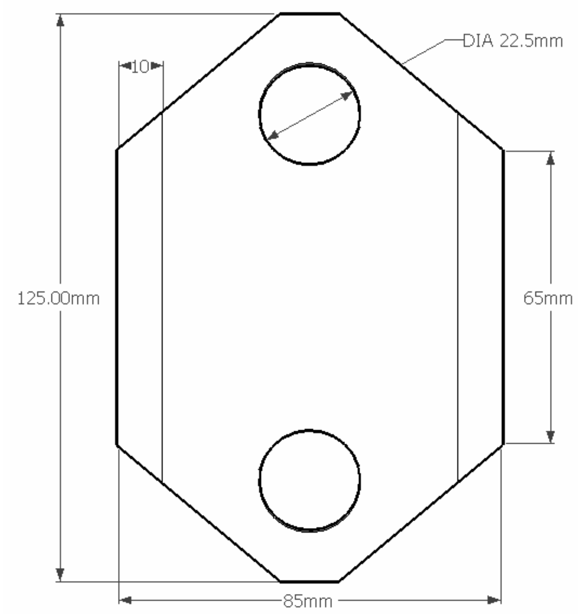

Figure 1. Bottom Frame.

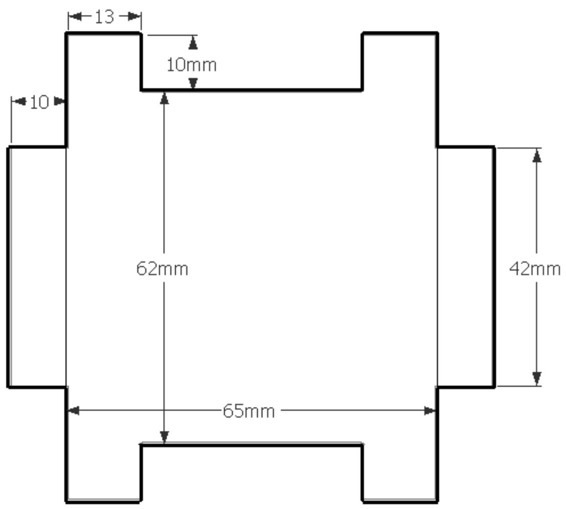

Figure 2. Top Frame.

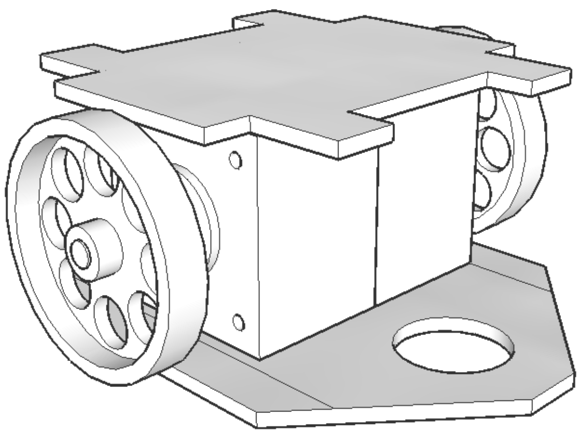

Figure 3. A perspective view of the robot's blueprint.

The stepper motors were controlled using Texas Instruments' SN75468. Its internal structure is shown in Fig. 4. The IC is an array of seven Darlington Pairs which act as switches to activate and deactivate individual windings of the stepper motors.

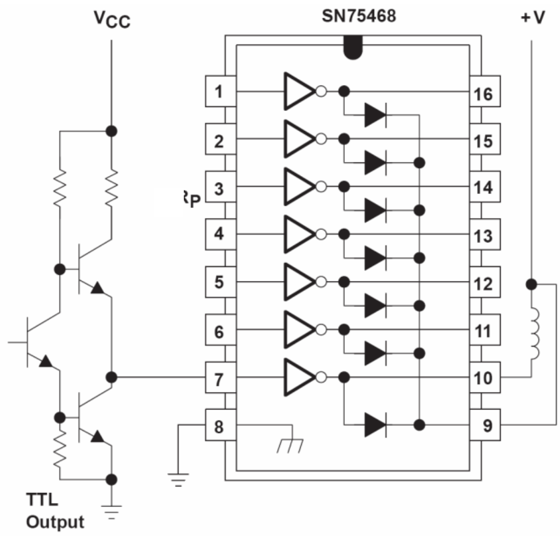

Figure 4. Controlling Stepper Motor Windings using the SN75468 IC.

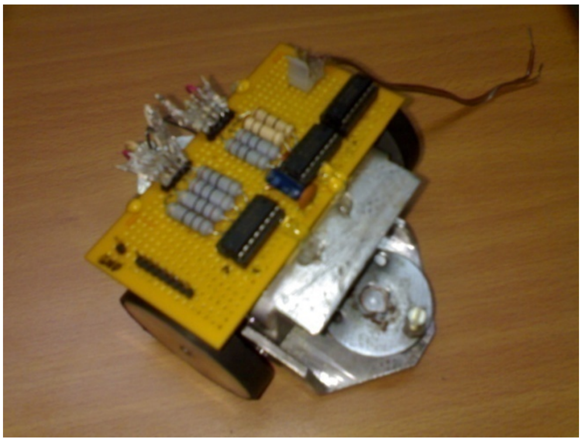

Figure 5. The Complete Robot

The input to this IC is given by the computer's Parallel Port. Our motors have four windings each. Hence each data bit of the parallel port controls an individual winding. This is advantageous as there is no need for an additional microcontroller or a USB or Serial interface IC at the robot end, thus reducing the cost.

The finished robot was slightly different than the planned blueprints. For example, the wheels were not drilled for convenience.

The circuit comprises of three SN75468 ICs with each winding being driven by two Darlington pairs in parallel for improved current capacity. Power resistors were added in series to the windings of the motors to limit the current. The circuit accepts input from the parallel port of a computer. The robot is powered by an external DC power supply. However it can also carry its own batteries for power.

IV. THE IMAGE ACQUISITION EQUIPMENT

The CCTV camera was mounted facing vertically downward using a stand we made. The CCTV camera was configured to capture images at a resolution of 800*600. Although the CCTV camera captures images in colour, we converted them to greyscale before doing any image processing.

The stand was designed such that the height of the CCTV camera could be adjusted. The CCTV camera's angle can also be adjusted with much ease. The stand also incorporates provisions for attaching CFL lamps for lighting purpose.

V. THE PROCESSING SYSTEM

All the computations required to collect data, process it and even control the robot were done by a desktop computer. All the programming was done using the MATLAB software. The final program was compiled into a standalone application. The robot can be controlled by the parallel port of a computer. We used a desktop computer for controlling the robot. The CCTV camera was connected to the computer using the USB Interface.

The code written in MATLAB can be compiled with code written in C++ to allow for online control of the whole system. The user, whether he/she is using the main computer or a computer connected to it, will have the ability to override the automated navigation and take control of the robot at any time. When the robot is under complete control of the computer, the user will be kept informed about the actions of the algorithm, for example, the path chosen for the robot, or the computer's view of the robot's surroundings.

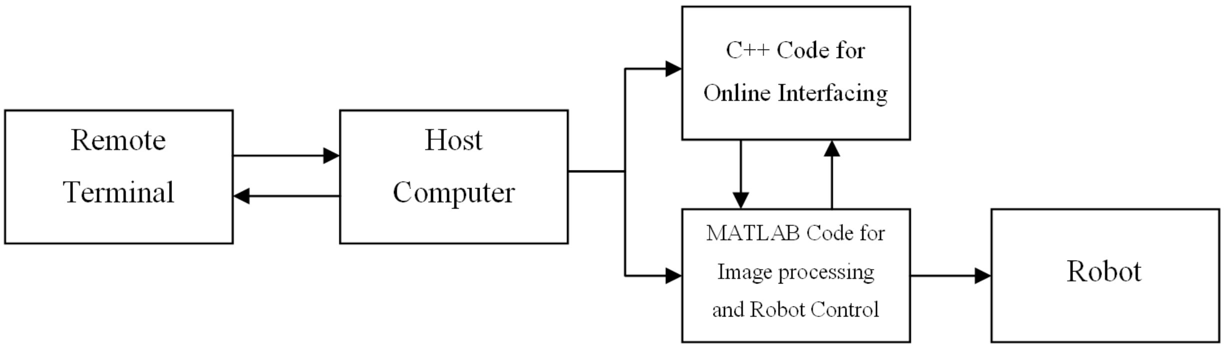

Figure 6. Block Diagram of the Proposed System for Online User Control.

VI. THE ALGORITHM

The algorithm we designed for this prototype was an obstacle avoidance algorithm. The computer obtains an image of the surroundings of the robot from the CCTV camera. It uses a learned map[3] for the robot's navigation. The obstacles are detected and a safe path for the robot to move to its destination is determined. Then the robot is run along the determined path. The objects are assumed to be stationary.

This algorithm was a cell-based approach at solving the path finding problem. It works by dividing the edge-detected image into squares roughly the size of the robot itself and categorizing the cells as filled or empty. Afterwards, the path is found by simply running a modified wall follower algorithm.

First, a binary image is obtained after edge detection[4],[5] as follows:

1. Grab an image from the CCTV camera.
2. Convert this image to greyscale form and store in "Image".
3. Apply the Wiener filter[6] to reduce noise.
4. Apply the Canny edge detection algorithm to obtain the "Binary Image".

Afterwards, this image is divided into cells. In our case we took the cell size as square of side 100 pixels. In this case, an 800*600 image is divided into 8*6 squares. An array called "Cell" is created that represents the state of each square as filled or empty. In this case, this array will be of dimensions 10*8 because bordering "filled" cells surrounding the original ones are created so that the later path finding algorithm will never wander beyond the 8*6 set of Cells. The algorithm is listed below:

1. Divide the whole image into squares of specified size.
2. Create a two dimensional matrix "Cell" whose members can take binary values. The size of this matrix is of dimensions 2 plus the number of cells. Initialize its bordering elements as 1 and the rest as 0.
3. Check each square if it contains any white pixels, its corresponding cell is marked as 1 i.e. filled.

Afterwards a dead end filler algorithm is executed. This algorithm simply traverses each cell of the array. If any cell surrounded by 3 filled cells is found, it is turned into a filled cell. This process is repeated until there are no more dead end cells present to fill.

If we were solving a maze, the dead end filler algorithm itself would have solved the entire maze. In this case however, it simply makes the path of the robot more direct.

Thus an array of cells approximating the layout of the whole image is created. A modified wall follower algorithm is used to find the path. The algorithm is listed below:

1. Starting from the origin, check if all the cells to the east are empty. If yes, move directly east to the target zone, else proceed to step 2.

2. Check if the cell to the left of the robot is empty, if yes, move to that cell and go to step 1 else proceed to step 3.
3. Check if the cell in front of the robot is empty, if yes, move to that cell and go to step 1 else proceed to step 4.
4. Check if the cell to the right of the robot is empty, if yes, move to that cell and go to step 1 else go back.

During the execution of the above algorithm, each motion is recorded into an array. This array contains the actions to be performed by the robot to reach the destination.

However, the array movement usually contains several consecutive forward movements. It's best to concatenate these movements to make the motion of the robot smooth. Once this is done, the robot is run along the resulting path.

Fig. 8, 9, 10, 11 depict the results of different parts of the algorithm in the sequence of the actions performed by the algorithm.

The path finding algorithm returns a path to be followed by the robot as sown in Table II. In this case, the starting point was located at second column of the second row in the cell array. The target zone was the second last column. A negative number –n represents forward motion by n cells. A positive number represents turns or backward motion. Here, 2 represents a left turn, 3 represents a right turn and 4 represents moving back.

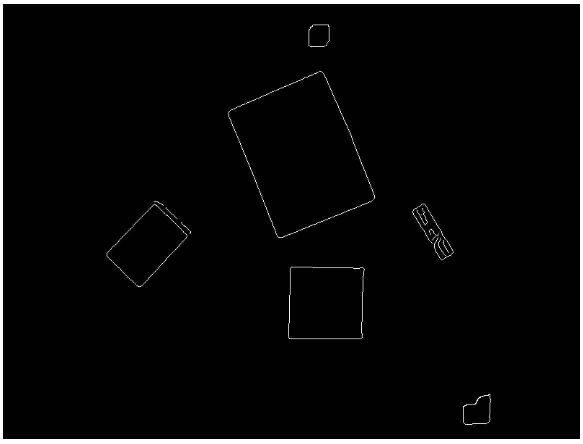

Figure 7. Binary Image obtained after edge detection.

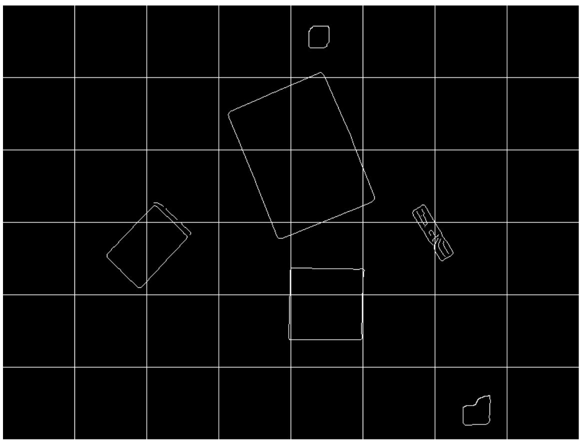

Figure 8. Edge Detected Image divided into cells.

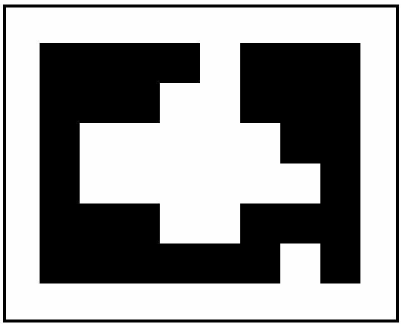

Figure 9. An image depicting the state of the cell array before filling the dead ends. White represents filled and black represents empty (safe) cells.

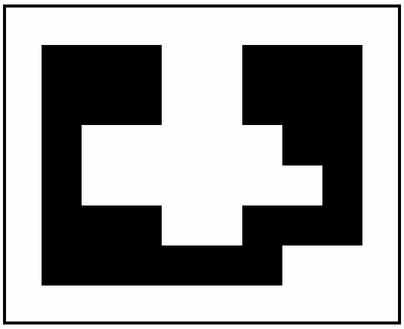

Figure 10. An image depicting the state of the cell array after filling the dead ends. White represents filled and black represents empty (safe) cells.

TABLE I.
THE STATE OF THE CELL ARRAY AFTER FILLING THE DEAD ENDS. 1 REPRESENTS FILLED AND 0 REPRESENTS EMPTY (SAFE) CELLS.

1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	0	0	0	1	1	0	0	0	1
1	0	1	1	1	1	1	0	0	1
1	0	1	1	1	1	1	1	0	1
1	0	0	0	1	1	0	0	0	1
1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1

Table II.
THE PATH TO BE FOLLOWED BY THE ROBOT.

-2	3	-1	3	-2	2	-3	2	-2

VII. CONCLUSION

This paper proposed the architecture of an online, cheaper, simpler and more dynamic robot for use in the industry by using a stationary CCTV camera as the primary sensing equipment and a computer as the main processing unit.

Although this prototype was developed as to demonstrate the proposed generalized approach for robots, it can also be used directly into several situations as listed below:

1. Fork Lifter Robot in factories. For picking up and placing objects. Automatic fork lifts have been made, but they only perform a set of pre-programmed operations, they cannot find objects and stack them, neither can they avoid obstacles.
2. Robots for finding and retrieving lost parts in factories.
3. Intelligent Vacuum Cleaner robots that will work cooperatively and divide the work amongst themselves.
4. A Parking Guide Robot that will guide any car coming to a parking facility to an empty lot. It will also assist the driver in backing out the car etc.

Although this prototype used the CCTV camera facing vertically downward, the same approach can be used for cameras viewing the surroundings at any angle when the angle is known. As most CCTV carry orientation controls using servo mechanism, their angle is always known. Thus the image obtained from these cameras can be transformed accordingly.

Moreover, the same system can control several robots at once and since they will all be controlled centrally, their behaviour can be collaborative e.g. by dividing a task amongst the robots, or planning the paths to avoid collision.

REFERENCES

[1] Gerd Hirzinger "Why Robots are just around the corner" *Pictures of the Future*,63, 2002.

[2] Rodney Brooks. "A hardware retargetable distributed layered architecture for mobile robot control." *IEEE Transactions on Robotics and Automation*, pp.106-110, May 1986.

[3] Peter Kampmann, Gunther Schmidt. "Indoor Navigation of Mobile Robots by Use of Learned Maps" *Information Processing in Autonomous Mobile Robots*, Springer Verlag, Berlin Heidelberg, New York, 1991.

[4] J. Canny, "A Computational Approach to Edge Detection" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986, pp. 679-698.

[5] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins "*Digital Image Processing Using Matlab*" 1st ed. Prentice Hall, 2004, pp. 379-393.

[6] J. S. Goldstein, I. S. Reed, and L. L. Scharf, .A multistage representation of the Wiener filter based on orthogonal projections, *IEEE Trans. Information Theory*, vol. 44, no. 7, pp. 2943.2959, November 1998.

AUTHORS

Akhil Piplani is a student at Department of Electronics and Communication, Jaypee Institute of Information Technology University, Noida, U.P. 210301, India.

Abhinav Gupta is senior lecturer at Department of Electronics and Communication, Jaypee Institute of Information Technology University, Noida, U.P., 201301, India.

Shailendra Mahani is a student at Department of Electronics and Communication, Jaypee Institute of Information Technology University, Noida, U.P. 201301, India.

Manuscript received 30 May 2007. Published as submitted by the authors.