

Line Correction for Robot Go-Forward using Image Detection Approach

Shih-Chang Hsia

National Yunlin University of Science and Technology,
Douliou, Taiwan
Email: hsia@yuntech.edu.tw

Wei-Siang Fu

National Yunlin University of Science and Technology,
Douliou, Taiwan
m10113215@yuntech.edu.tw

Abstract - In this paper, we present a line correction method for robot moving control based on image detection approach. By statistics of horizontal vector, the line deviation can be estimated, and then to control the speed of motor to trim the moving direction. For real-time control requirement, we presented a fast estimation algorithm based on binary successively approximate approach, to reduce the computation complexity about 72.3%. This method is successfully implemented on a real wheel robot system, which can perform line correction in real-time using camera imaging processing.
Keywords-robot, motion search, line correction, image

I. INTRODUCTION

The use of Robots will be more and more in the future. There are many kinds of robots for various applications. Generally, there are two types for robot classifications according to motion features: one is moving-able robot, such as human-like and wheel-based robot, and the other is fixed on worktable, such as machine arm. In the former, the applications will be to close to people living. In the latter, most of applications are in industrial manufactures. The moving-able robot can be interactive with humans, which require intelligent functions by sensing environment. For robot control, we need many kinds of sensors, such as (1) camera : like human eyes, to detection environment by image recognition; (2) microphone: like human ears, to listen environment sound; (3) ultrasonic sensor: can detect the distance between robot and obstacles.

In this study, we proposed a smart robot for house patrol using wheel-based design. This robot can monitor house condition and send the videos to remote devices. such as smart cell phones. For cover wide monitoring area, the robot uses wheels to move to anywhere. However, we face a main problem: the robot cannot forward go as providing the same speed to all motors of wheels. The robot may loss control, sometimes turn right and sometimes turn left dependent on various floors. Friction of wheels to floor is different in practical, causing the motor driving power for wheel is different and the moving direction of robot is not predicted when commanding to robot. To solve this problem, we proposed vision-based to correction the error of deviation. When finding the robot moving direction is not

predicted, the correct algorithm can compute suitable parameters and send to motor to control the speed of left and right wheels.

The rest of this paper is organized as follows. The proposed linear correction algorithm for robot is presented in Section II. The implementation and experiments are described in Section III. The conclusions are marked in Section IV.

II. PROPOSED ALGORITHM

In this paper, we employ image processing approach to localize the robot moving. First, the sum of absolute difference (SAD) is computed along horizontal direction, which can be given by

$$SAD_X = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f_i(q+i, p+j) - f_{i-1}(q+i, p+j)|, \quad (1)$$

, where f_i and f_{i-1} are the current and the previous n^{th} frame respectively, (q, p) is the position of macro block, and N is the block size. Compute horizontal vector from $-k$ to $+k$, and to find a minimum SAD_X and record the current vector, as shown in Fig. 1.

Figure 2 shows the processing flowchart of the proposed line correction for robot moving. The camera is located on the head of robot. The image is sampled and sent to flat compute using USB interface. The frame is first stored to memory. Next, we can compute SAD_X using (1) by the current frame and the previous frame within a $N \times N$ block. From full-searching $-k$ to $+k$ vectors, one can find a minimum SAD_X that is the horizontal motion vector. If the SAD_X is larger than the threshold $th1$, implying that the robot may be not on the line. Then the SAD_X value and its relative motion vector are saved to register. When a positive motion vector is found, the data is accumulated to Reg1, otherwise to Reg2. The next block is processed by using the same flow. When all blocks had been processed for one frame, we can find a maximum SAD_X value between Reg1 and Reg 2. Then the motion vector from the maximum SAD_X is averaged as a control parameter for controlling motor of wheel.

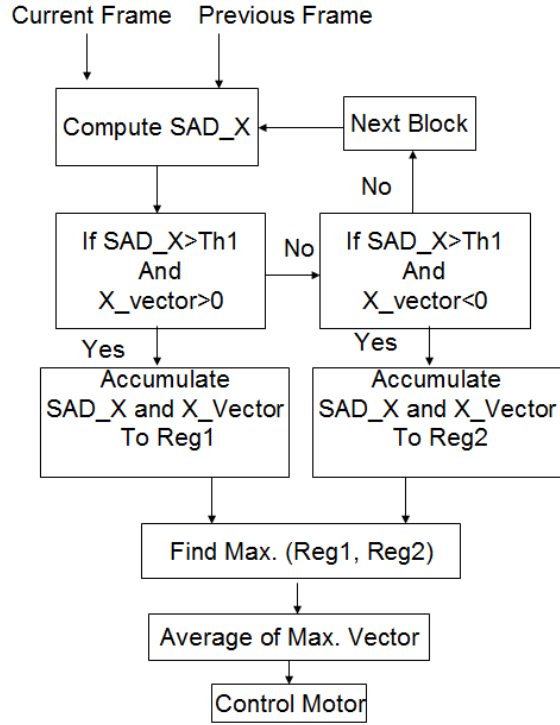


Figure 2 The processing flow of line correction algorithm.

If the motion vector is positive, the deviation direction of robot may turn right. Such a case, we speed-up the motor of the right wheel and speed-down the motor of the left wheel. Then checking the motion vector of next frame, we continue to trim until to the all over motion vector is less than the threshold. Oppositely, if the motion vector is negative, the deviation direction of robot may turn left. Then, we speed-up the motor of the left wheel and speed-down the motor of the right wheel to force the robot to right for line correction.

For robot direction control, we need to control the moving direction of robot immediately, so a real-time issue is very important. From (1), if $N=16$, we require 256 subtractions and 255 additions for computing one SAD value. When $k=16$, to find one motion vector of one block, we require 4352 subtractions and 4335 additions. The computational complexity is too for real-time robot control. To reduce computational complexity, we present a fast algorithm to find the horizontal vector for robot control based on binary splitting approach. First, we compute the SAD_X value of the points $-k/2$, 0 , $+k/2$, and then to find the minimum SAD_X among three points, as shown in Fig. 3(a). If the minimum SAD_X is located at the points $-k/2$, then the central point is moved to $-k/2$. We compute SAD_X of the points $(-k/4+-k/2)$ and $(-k/2+k/4)$, and to find the minimum SAD_X among $(-k/4+-k/2)$, $-k/2$, $(-k/2+k/4)$ points in the second step, as shown in Fig. 3(b). By using

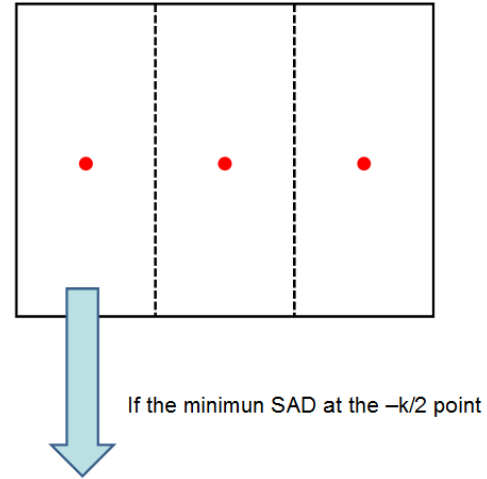


Figure 3 (a) Compute SAD_X of $-k/2$, 0 , $+k/2$ points in the first step.

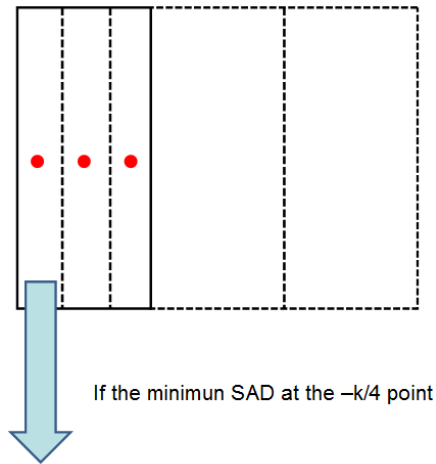


Figure 3 (b) Check the minimum SAD_X among $(-k/4+-k/2)$, $-k/2$, $(-k/2+k/4)$ points in the second step.

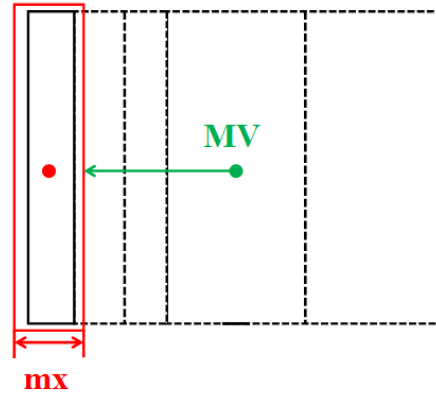


Figure 3 (c) Find the horizontal vector in the last step.

this processing flow, we can find the final vector at the last step, where the differential vector of last three points should be one, as shown in Fig. 3(c).

For example, if the real vector is at -15, first (-8,0,+8) points are checked in the first step. The minimum SAD_X should be at the -8 vector. Then, the points of (-12, -8, -4) are checked in the second step, where we only compute SAD_X of -12 and -4 vectors and SAD_X of -8 had been calculated in the first step. The minimum SAD_X would be at the -12 point. Then the points of (-14, -12, -10) are checked in the three step and find the minimum SAD_X at -14. Finally, we calculate the SAD_X of -15 and -13 points, and find the minimum SAD_X among -15,-14 and -13. The differential vector is one that is confirmed at the last step, and we can find the best vector at -15. Using this binary successively approximate method, the computation can be greatly reduced. In original full-search, we need to compute SAD_X of 33 vectors when $k=16$. In this approach, we only compute 3, 2, 2, 2 points in the first, second, three and last step respectively. In total, only 9 points are checked. The computation can save $(1-9/33)=72.7\%$.

III. IMPLEMENTATION ON WHEEL ROBOT

The robot system is constructed by using two wheels that is controlled by motor, and one all-direction wheel for prop up the robot that is fixed at the tail. We built three ultrasonic modules to calculate the distance from robot to obstacles. Figure 4 shows the function block of the robot control system. The control kernel used a micro-processor with a single chip. The two external driver chips are used to drive two motors to control two wheels for robot moving. A lithium battery is used to supply the power of motor, and using buck chip to drop the voltage for micro processor. Two digital SR motor is used at the head of robot that can be directly controlled for up/down and left/right for camera using micro-processor. We used one blue-tooth module for communicating with computer.

The flat computer is built in on the robot for communication, computation and human interface. The camera of head sent the video signals to flat computer, and C-programming is used to develop the line correction algorithm based on robot platform. Using the proposed fast algorithm, we can process 3~5 frames to find the motion vector. By statistics of horizontal vector, the deviation direction can be estimated at once. The flat-computer sent the estimated parameters to micro-processor through blue tooth. According to the estimated parameters, the micro-processor used PWM (pulse width modulation) to control the speed of right-wheel and left-wheel motors and to change the moving direction. When finding the moving deviation is not on line, the micro-processor can command the motor to correct it to line.

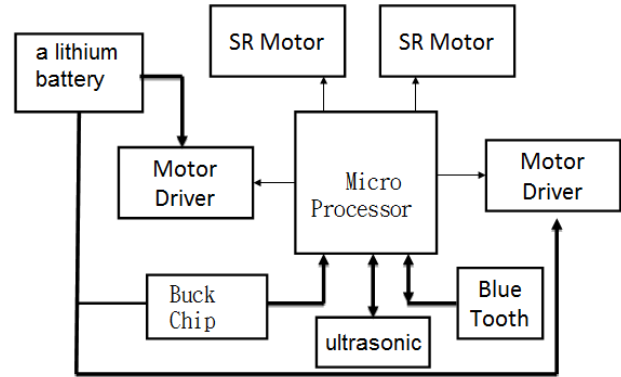


Figure 4 The robot control system.

Now we show practical results in experiments. Figure 5(a) and (b) shown the results without line correction at the 10 frame and 30 frame respectively, where two driving motors used the same speed. Clearly, the robot appears to turn right deviation. If not to correct, the robot may turn in cycle.



(a)



(b)

Figure 5 The non-correct robot moving (a) at the 10 frame; (b) at the 30 frame (turn right deviation).

Next, we used our proposed line correction method for robot. The camera sampled the videos to computer processing and to control motor immediately. Figure 6 (a) and (b) shown the moving direction at the 10 frame and the 30 frame respectively. From experiments, the result shown that the robot can move on line after our line correction.



(a)



(b)

Figure 6 The proposed correct method for robot moving (a) at the 10 frame; (b) at the 30 frame (direction correct).

V. CONCLUSIONS

In this paper, we study the line correction algorithm for robot moving control. Using image processing approach, we compute the over-all horizontal vectors to estimate the deviation direction in statistics. To improve the control speed for real-time purpose, the fast binary successively approximate approach is proposed. The micro-processor can change the motor speed 3~5 times per second. Experimental results shown that the robot can move on line after our correction processing. The method is very value for robot navigation, patrol and auto localization system.

REFERENCES

- [1] K. Okuyama, T. Kawasaki, and V. Kroumov "Localization and Position Correction for Mobile Robot Using Artificial Visual Landmarks" IEEE International Conference on Advanced Mechatronic Systems , August 11-13, 2011, pp. 414-418.
- [2] A. Ryberg, M. Ericsson, A.-K. Christiansson, K. Eriksson, J. Nilsson and M. Larsson, "Stereo Vision for Path Correction in Off-Line Programmed Robot Welding", IEEE Conf. Industrial Technology, 2010, pp. 1700-1705.
- [3] D. Stavrou, D. G. Eliades, C. G. Panayiotou and M. Polycarpou, " A Path Correction Module for Two-Wheeled Service Robots Under Actuator Faults", 21st Mediterranean Conference on Control & Automation (MED), June, 2013, pp. 1119-1126.
- [4] Peter Lim, Liyanage C. De Silva, Liu Jing, and Li Li Ling, "Multimodal Approach to Human-Face Detection and Tracking" *IEEE Trans. INDUSTRIAL ELECTRONICS*, VOL. 55, NO. 3, PP. 1385 – 1393, MARCH 2008.
- [5] Liyuan Li, Shuicheng Yan, Xinguo Yu, Yeow Kee Tan, and Haizhou Li, "Robust Multiperson Detection and Tracking for Mobile Service and Social Robots", *IEEE Trans. SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, VOL. 42, NO. 5, PP. 1398 – 1412, OCT. 2012.