

Adjustment of Viterbi Algorithm for Line Following Robots

Grzegorz Matczak and Przemysław Mazurek

Abstract Line following robots are used in numerous application areas. The tracking of weak line is challenging, especially if SNR is high, so application of Track–Before–Detect algorithm is necessary. The Viterbi algorithm is assumed in this paper and the possibilities of optimization are considered. Two metric are applied in Monte Carlo tests—the direct metric and proposed boundary metric. The optimization of Viterbi algorithm is based on non single row movements of moving window.

Keywords Track-Before-Detect · Tracking · Viterbi algorithm · Line following robots

1 Introduction

Robot navigation could be based on satellite navigation (GPS), inertial navigation or vision systems. Vision based navigation requires sophisticated algorithms for general case, but the simplification for well known or defined working area is possible. Line following robots use intentionally added line that defines possible routes. Many line following robots are proposed and used, because vision system could be very simple. Simplest systems uses two light reflective sensors for the measurement of reflected light from the line and the background. Fixed value for both measurements is obtained if the sensor is over the line. Different values occurs if the sensor is misaligned and the error could be used by the control algorithm. Rapid movements and high velocities require more advanced sensors, so linear light sensors are applied also. Forward looking cameras (Fig. 1) allow the estimation of line before robot

G. Matczak (✉) · P. Mazurek

Department of Signal Processing and Multimedia Engineering, West Pomeranian University of Technology Szczecin, 26. Kwietnia 10 St., 71126 Szczecin, Poland
e-mail: grzegorz.matczak@gmail.com

P. Mazurek

e-mail: przemyslaw.mazurek@zut.edu.pl

Fig. 1 Schematic of line following robot with forward looking camera

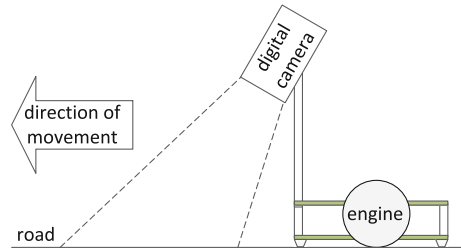
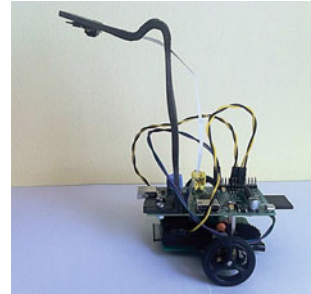


Fig. 2 Example line following robot with forward looking camera



(Fig. 2) at far distance also and this is the best solution, but requires sophisticated algorithms. Systems with artificially added lines are simple to design and the desired high background to line contrast could be achieved.

There are numerous application where the line is natural or artificial but significantly degraded. The quality of the line is low in such cases, so the contrast between the background and the line is not sufficient for the application of simple vision systems and algorithms. This paper assumes such cases, that are important for numerous applications of line following robots.

1.1 Related Works

Navigation using lines, that are highly deteriorated are considered in numerous Lane Departure Warning (LDW) systems. The traffic safety could be improved if LDW system is used [14]. The very interesting research area is the combined navigation and inspection of power lines by UAVs [2]. There are numerous application of line following robots related to the harvesting, trash compacting [12].

Line estimation could be improved by the application of Track-Before-Detect algorithms. Such class of tracking algorithm [1, 13] allows the line estimation for $SNR < 1$ cases also. Viterbi algorithm [15] for line following robots is proposed

in [8]. Lines that are noise pattern only, could be processed by the approach considered in [10]. Directional filtering of lines improves estimation also [7]. Local approach for the detection of disturbed lines is considered in [4] for example.

1.2 Content of the Paper

Two metrics for quality of estimation are considered in Sect. 2 (direct and new proposed boundary metric). The selection of metric influences the results and it is shown in Sect. 2. The optimization of Viterbi algorithm could be based on different parameters and two of them are selected and considered in Sect. 3 due to mobile robot applications. Monte Carlo tests are provided in Sect. 4. The results and discussion are provided in Sect. 4. Final conclusions are described in Sect. 5.

2 Metrics for Horizontal Error

The direct metric (E_i) for the horizontal error value could be defined as a difference between true X_i and estimated \hat{X}_i positions:

$$E_i = X_i - \hat{X}_i. \quad (1)$$

Alternative approaches are possible and another metric is proposed (E_i^B) that considers width of line, so left (X_i^L) and right (X_i^R) boundaries are used. The error value in horizontal direction is typically non-zero for direct metric, but the error value should be zero if the estimated line position is located between line boundaries:

$$E_i^B = \begin{cases} 0 & : X_i^L \leq \hat{X}_i \leq X_i^R \\ X_i^L - \hat{X}_i & : \hat{X}_i \leq X_i^L \\ \hat{X}_i - X_i^R & : X_i^R \leq \hat{X}_i \end{cases}. \quad (2)$$

Estimated line position (Fig. 3 left), with pixel accuracy, that is inside boundaries gives zero value error for direct metric. The application of boundary metric allows the acceptance of all position between boundaries (Fig. 3 right).

New boundary metric is proposed, because the selection of metric has impact on results. The example of line tracking for wide line is presented in Fig. 4 and direct metric and boundary metrics are compared. The cumulative error value is significantly reduced which shows impact of the metric. The results of Monte Carlo test for variable noise are shown in Fig. 5 and the difference between both metrics is well visible.

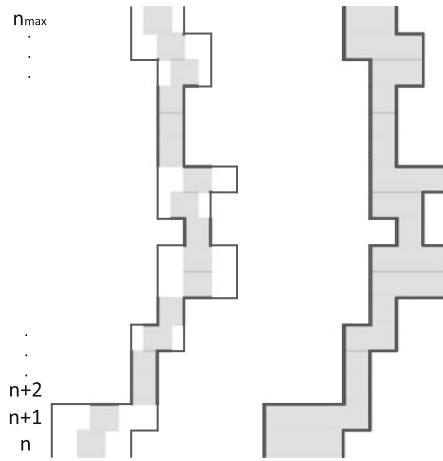


Fig. 3 Direct (*left*) and boundary metrics (*right*) regions of zero value error (*gray*)

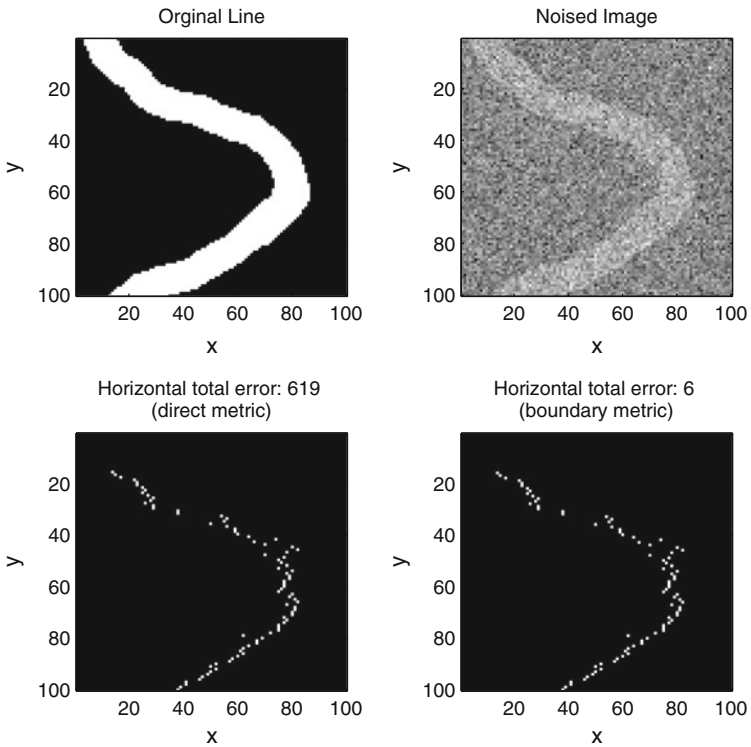


Fig. 4 Example results for std. dev. = 1.0

3 Adjustment of Viterbi Algorithm for Mobile Robots

Moving windows approach has a high degree of similarity to driving mobile robot with digital camera looking down on route. Mobile robots very often moves at different speed and directions. Windows approach ensure possibility of moving mobile robot simulation in the same direction with different speed.

Windows approach from previous publications [7, 8, 10] defines shifting windows in image. Window has dimensions width as image width and window height as *depth* parameter. Window in the image was started from $n = 1$ row and finish on $n = n_{max}$, where n_{max} is known from value of *depth* parameter. Next processing of window starts from $n = 2$ row and finish on $n = n_{max} + 1$ row. The shift of window is equal one pixel in this case. This is the smallest possible move of the window in simulation. Added ability to change value of shifted window contributed to model mobile robot speed. If we parametrize this adding part we get $(n + step)$, where the *step* is mentioned parameter of shifted window. Markings shown in Fig. 5.

The selection of *depth* and *step* parameters depends on several factors. First is resolution of image. The image from digital camera can be in different resolution. When resolution is low, the computation is fast but with less accuracy. When resolution increases, the accuracy is improved and the number of computation is growing. When the same image is captured with differs resolution, the depth parameter is various, because for higher resolution gained greater number of pixel. Second factor is height of the digital camera relative to ground, because when camera is on higher position it can register more route. Last factor is *FPS* (Frames Per Second) value of

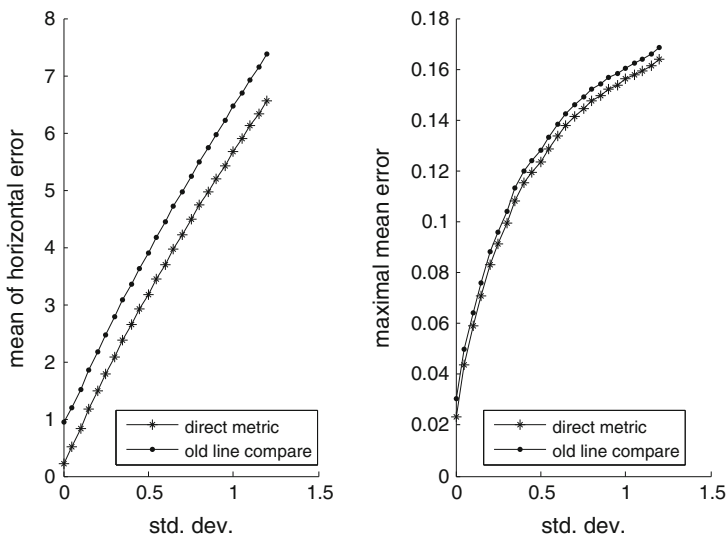


Fig. 5 Mean of horizontal error (left) and maximal mean error (right). Monte Carlo test for std. dev. = 0.13 and line direction change probability 0.25

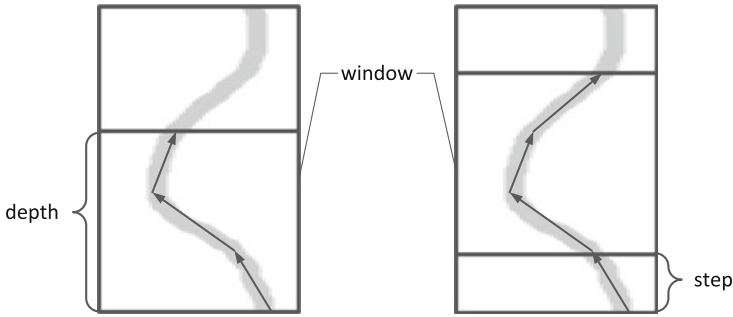


Fig. 6 Shifted window approach

digital camera. Higher number *FPS* provided better accuracy with following line, because faster moving of mobile robot can be register (Fig. 6).

4 Results and Discussion

The same set of generated line was made on the image with 200×60 pixel resolution. After generated line, disturbing lines with standard deviation from Gaussian noise were added. Next to the image Gaussian noise was added. Each set has 300 test scenario for specified Gaussian noise in std. dev. range (0–1.3) every 0.05 value.

Two values of parameters were studied in this paper: $depth = (16, 32)$, and three value of parameter $step = (1, 0.25 \cdot depth, 0.5 \cdot depth)$. For parameter $depth = 16$ and $step = (1, 4, 8)$, results are shown in Fig. 7. The results for parameter $depth = 32$ and $step = (1, 8, 16)$ are shown in Fig. 8.

The change of parameter $step$ does not aggravate the algorithm, so algorithm will work with high speed of mobile robot. The result is better for higher noise level. The number of computation changes with step parameter. The algorithm finds faster for higher step parameter. The comparison between different depth parameter (Figs. 7 and 8) shown that parameter have small impact on the results.

The algorithm is resistant to change speed of mobile robot that is related to $depth$ parameter. If we want our robot to move quickly we need to have as much as possible scope of the road. It can be achieved in several ways: by increasing the field-of-view of camera, or by increasing of height from the ground. The camera with more *FPS* could be applied for the better accuracy.

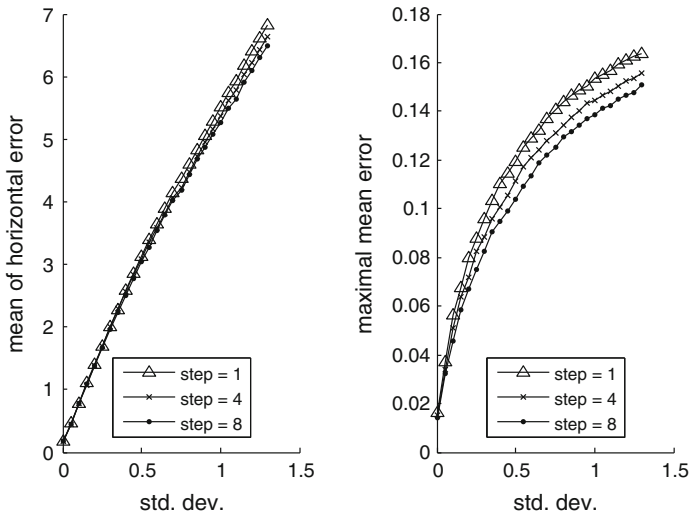


Fig. 7 Mean error and maximal mean error for different *step* parameter, (Monte Carlo test for std. dev. and *line direction* change probability 0.25 and *depth* = 16)

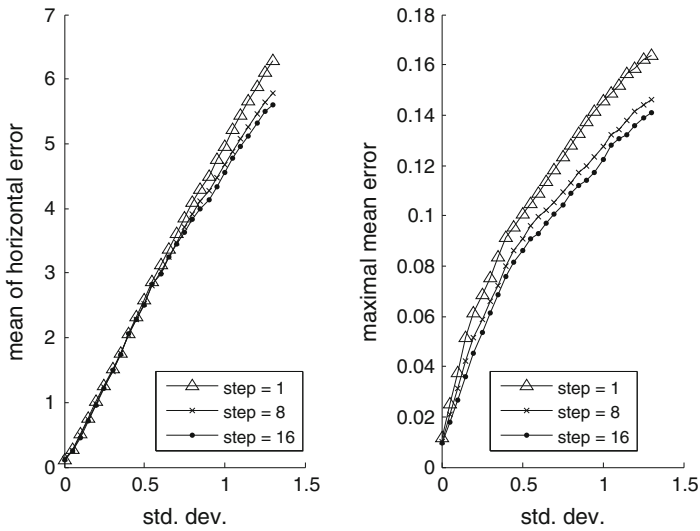


Fig. 8 Mean error and maximal mean error for different *step* parameter, (Monte Carlo test for std. dev. and *line direction* change probability 0.25 and *depth* = 32)

5 Conclusions

The computational cost of considered algorithm is acceptable for modern micro-controllers for low-resolution images. Proposed approach allows the reduction of computation cost by the application of larger step of moving window. High resolution images need TBD processing using multiple processors [9] or GPGPUs [5, 6].

Presented example assumes additive Gaussian noise disturbance. The removal of non-line object from the image could be based on concept presented in [3]. Texture related disturbances could be also identified [11].

References

1. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House (1999)
2. Chen, Z., Ellis, T.: Automatic lane detection from vehicle motion trajectories. In: Workshop on Vehicle Retrieval in Surveillance (VRS) in conjunction with 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 466–471 (2013)
3. Frejlichowski, D., Gościewska, K., Forczmański, P., Nowosielski, A., Hofman, R.: The removal of false detections from foreground regions extracted using adaptive background modelling for a visual surveillance system. Computer Information Systems and Industrial Management. Lecture Notes in Computer Science, vol. 8104, pp. 253–264. Springer, Berlin (2013)
4. Marchewka, A.: Crack detection on asphalt surface image using local minimum analysis. Adv. Intel. Soft Comput. **84**, 353–359 (2010)
5. Mazurek, P.: Optimization of bayesian track-before-detect algorithms for GPGPUs implementations. Electr. Rev. R. **86**(7), 187–189 (2010)
6. Mazurek, P.: Code reordering using local random extraction and insertion (LREI) operator for GPGPU-based track-before-detect systems. Soft Comput. **18**(6), 1095–1106 (2013)
7. Mazurek, P.: Directional filter and the viterbi algorithm for line following robots, Computer Vision and Graphics, ICCVG 2014, LNCS, vol. 8671, pp. 428–435. Springer (2014)
8. Mazurek, P.: Line estimation using the viterbi algorithm and track-before-detect approach for line following mobile robots. pp. 788–793 (2014)
9. Mazurek, P.: Parallel distributed downsampled spatio-temporal track-before-detect algorithm. pp. 119–124 (2014)
10. Mazurek, P.: Viterbi algorithm for noise line following robots. Adv. Intel. Syst. Comput. AISC **313**, 111–118 (2015)
11. Okarma, K., Frejlichowski, D., Czapiewski, P., Forczmański, P., Hofman, R.: Similarity estimation of textile materials based on image quality assessment methods. Lect. Notes Comput. Sci. **8671**, 478–485 (2014)
12. Ollis, M.: Perception Algorithms for a Harvesting Robot. CMU-RI-TR-97-43, Carnegie Mellon University (1997)
13. Stone, L., Barlow, C., Corwin, T.: Bayesian Multiple Target Tracking. Artech House (1999)
14. Taubel, G., Yang, J.S.: A lane departure warning system based on the integration of the optical flow and Hough transform methods. In: 2013 10th IEEE International Conference on Control and Automation (ICCA) Hangzhou, China, June 12–14, pp. 1352–1357 (2013)
15. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory **13**(2), 260–269 (1967)