# Directional Filter and the Viterbi Algorithm for Line Following Robots

Przemysław Mazurek

West–Pomeranian University of Technology, Szczecin
Department of Signal Processing and Multimedia Engineering
26–Kwietnia 10 St. 71126 Szczecin, Poland
przemyslaw.mazurek@zut.edu.pl

**Abstract.** The line estimation algorithm dedicated to line following robots is proposed in this paper. The line estimation is based on the Viterbi algorithm and directional filtering using moving average filter. Two cases of system are compared using Monte Carlo approach – with proposed directional filter and without this filter. The performance is measured using comparison of cumulative errors for horizontal direction. The results shows 20% improvements for Gaussian noise standard deviation $0.3 - 0.9$ range, for proposed solution in comparison to the Viterbi algorithm alone approach.

**Keywords:** Dynamic Programming, Tracking, Image Processing, Viterbi Algorithm.

## 1 Introduction

Line following robots are applied for components delivery in manufactures [1], especially. They are also applied in many contests in robotics and for educational purposes, because they could be very simple and cheap. The line following robot uses linear sensor (1D) or camera (2D) and very high line–to–background contrast is assumed typically. Even two light sensors with simple logic [2] for the line recognition could be applied for such intentional assumption. Real application of line following robots requires more robust image processing algorithms and pattern recognition techniques. The line could be deteriorated and lighting conditions could be variable. Locally variable lighting conditions, and noises related to the deteriorated background and real line as well as false lines are the sources of low quality of acquired image. The positive signal of line and dark background are assumed (or opposite mapping). The application of the camera allows the line estimation (tracking) but low–quality image requires robust image processing and tracking algorithms. The line width could be variable due to deterioration as well as low contrast that are shown in example in Section 4. Both problems are considered in this paper and the solution is provided.

L.J. Chmielewski et al. (Eds.): ICCVG 2014, LNCS 8671, pp. 428–435, 2014.

### 1.1 Related Works

The idea of line following robots is quite old [3]. The area of applications related to harvesting and similar task, where the line is not specified directly, is considered in [4]. Hough Transform approach for highly deteriorated line is considered in [5] for the road lane estimation as well as in agricultural application [6]. The application of LIDAR for the acquisition of 3D structures for line estimation is considered in [7]. The application of Monte Carlo sampling of image space is considered in [8].

### 1.2 Content and Contribution of the Paper

The problem of the line tracking with variable width and very–low contrast at the edge or below of edge of visibility by human is considered in this paper. Such assumption requires very robust algorithm, that will work very well even if the line or lighting conditions are significantly better. It allows the application of the proposed algorithm for scenarios, where the line is not intentionally added, so image existing line could be used, also. The line could be hidden in the image noise and the line width could be variable, due to deterioration. Multiple (false) lines that are parallel to the correct line may be image disturbance sources also.

In this paper the Viterbi algorithm [9] for line following robots is applied and briefly described in Section 2. The application of the directional filtering for image preprocessing is proposed in Section 3. The boundary possibilities of the line estimation are estimated using numerical experiments based on Monte Carlo approach in Section 4. Such assumptions allow the testing of many cases that are not available in small real image databases. The discussion is provided in Section 5 and the final conclusions are formulated in Section 6.

## 2 The Viterbi Algorithm for Line Estimation

The Viterbi algorithm is the dynamic programming algorithm that could be applied for the line tracking. Acquired image is analyzed, starting from the bottom row. The pixels may correspond to the nodes (states) of the trellis. Appropriate transitions between nodes of two following rows are related to the possible transitions. This is defined by the Markov transition matrix and probabilities of transition could be assigned. Five transitions with equal probabilities in this paper (Fig. 1) are assumed. The number of transitions depends on the line model.

The selection of the best path in the trellis corresponding to the most probable line is based on the local and global choice. Tracking process starts from the first (bottom) row ($n = 1$) and the local costs $d_{n,x}^{n+1,x}$ are computed for all paths allowed by trellis to the second row ($n = 2$), where $x$ is the position in horizontal direction. The local cost could be defined as a sum of the pixel values and highest values correspond to the possible line and should be preferred (local choice). The local cost for particular local path is added to the node value $V$ of the previous row $n$, and projected to the next row $n + 1$:

$$V_{n+1,x} = \max\left(V_{n,x+g} + d_{n,x+g}^{n+1,x}\right), \tag{1}$$

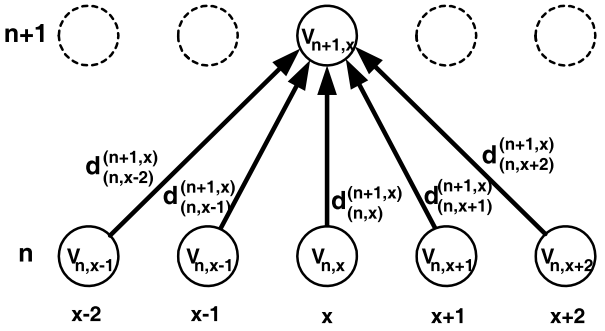


**Fig. 1.** Local paths in example trellis

where the transitions are defined using the following set:

$$g \in \{-2, -1, 0, +1, +2\}, \tag{2}$$

and the starting condition is:

$$V_{n=1,.} = 0 \tag{3}$$

The selection of the best path to the particular node is preserved additionally:

$$L_n^{n+1,x} = \arg\max_g \left( V_{n,x+g} + d_{n,x+g}^{n+1,x} \right), \tag{4}$$

where $L$ denotes local transition. The projection of the values is executed $n_{max}$ times. The depth of the projection ($n_{max}$) influences the estimation results, but this it is not considered in this paper.

After reaching of the $n_{max}$ row in first phase (forward phase), the second phase is started (backward phase). The maximal value $P_{opt}$ for $n_{max}$ selects the most probable path, starting from the last row:

$$P_{opt} = \max \left( V_{n=n_{max},.} \right), \tag{5}$$

that selects $x_{n=n_{max}}$ node:

$$x_{n=n_{max}} = \max_x \left( V_{n=n_{max},x} \right). \tag{6}$$

The following recursive formula finds path in trellis (backward):

$$x_{n-1} = x_n + L_{n-1}^{n,x}, \tag{7}$$

for successively decremented row numbers:

$$n = n_{max}, \cdots, 2. \tag{8}$$

The transition between first and second row for specific starting point $x_1$ is the result of the Viterbi algorithm - part of the estimated trajectory. Overall

process is repeated using moving window approach for the input image. Obtained estimation result of the Viterbi algorithm could be used in next processing steps, but it is not recommended. First forward–backward calculation could give wrong result and may propagate wrong result to next calculations if first result is reused in next computations. This property is related to the typical problems of initialization of recursive processing algorithms.

## 3 Directional Filtering for Preprocessing

Image processing of acquired image could improve performance of tracking algorithm and is allowed for the TBD (Track–Before–Detect) approach [10,11]. Proposed filtering technique is based on directional MA (Moving Average) filter. This is low computation cost FIR (Finite Impulse Response) filter, that is applied in vertical direction. The overall schematic of the proposed system is shown in Fig. 2.

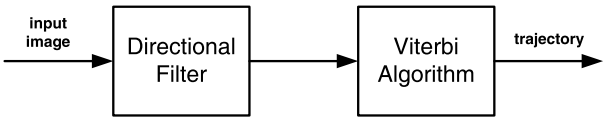


**Fig. 2.** Schematic of line tracking system with directional filtering

The directional MA filter uses simple formula:

$$Y(x,y) = \frac{1}{N}\sum_{i=0}^{N-1} X(x,y+i), \tag{9}$$

where $N-1$ is the filter order applied for vertical dimension ($y$) only. The $X$ and $Y$ variables are input and output images respectively. Such filter is dedicated to the line that is vertical or slightly sloped. General case of line with sharp turns is not considered is this paper. It is sufficient for many practical cases, but not for the specific robot following contests where the line could change direction in any way.

The aim of the filter application is the noise suppression [12], with reduced influence on the vertical edges, so blurring in horizontal direction is not recommended. This suppression is mostly independent on line and the background. The idea of directional filtering is the basis of the TBD systems based on e.g. velocity filters and local Hough Transform [13].

## 4 Results

An example input image and result of processing for directional filtering are shown in Fig. 3.

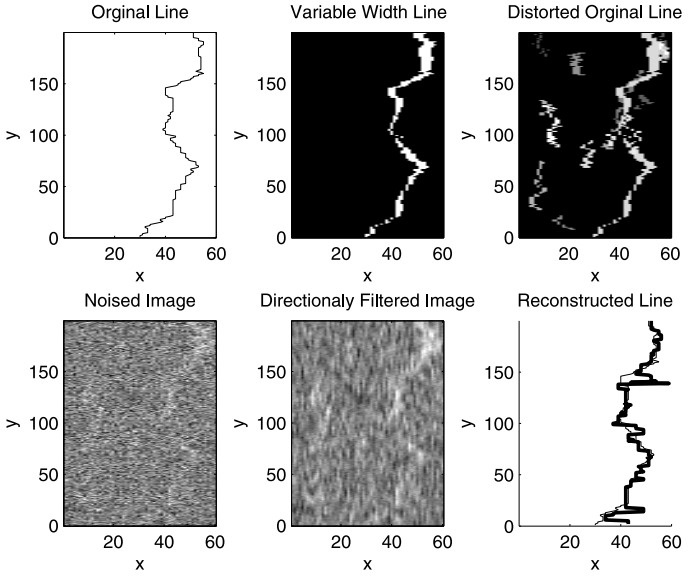


**Fig. 3.** Example results for specific image – high standard deviation 1.2

Monte Carlo approach is applied for the performance analysis under heavy distortions (multiple additional lines with similar spatial properties). Two systems are compared: without directional filtering and with directional filtering, for exactly the same input images. Monte Carlo tests uses 300 testing scenarios for particular standard deviation ($std$) and filter length ($N$). The $n_{max} = 60$, line width from $\langle 1 - 5 \rangle$ pixel range (probability of direction change is 0.1), line direction change with 0.25 probability are assumed for Markov g–set. Line length in $y$ direction is $L_y = 200$. Cumulative errors ($Cerr$) for both cases could be compared for the performance measurements. The pair of measurements $\{Cerr_{withfilter}, Cerr_{withoutfilter}\}$ marked by the point bellow line with slope 45° (equal error boundary) has better estimation result for system with directional filtering, and the point above this line has better estimation result for system without directional filtering (Fig. 4). The slope of regression line for multiple tests could be criterion of improvement for different filter orders.

The regression line $E_{withfilter} = f\left(E_{withoutfilter}\right)$ could be modeled using the following formula:

$$E_{withfilter} = a_1 E_{withoutfilter} + a_0, \tag{10}$$

and $a_0$ is very low value and could be omitted in regression fit. The values of $a_1 < 1$ are indicators of better performance of system with directional filtering. Mean error $\bar{E}$ for position in horizontal direction is:

$$\bar{E}_{withfilter} = a_1 \bar{E}_{withoutfilter} = E_{withfilter}/L_y. \tag{11}$$

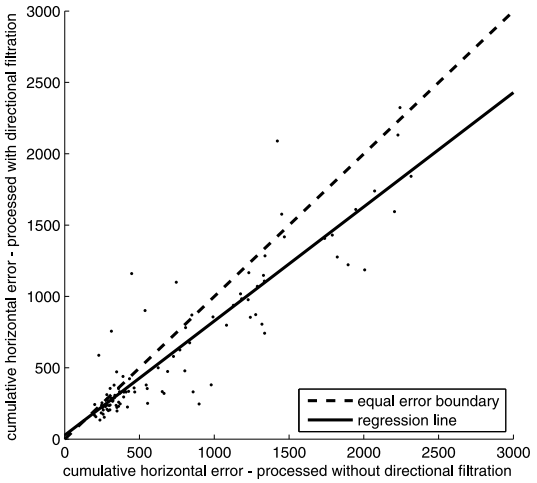


**Fig. 4.** Example results for Monte Carlo test for standard deviation 0.8

The highest value is about $2500/L_y = 12.5$ pixel and typical value is about $500/L_y = 2.5$ pixel for cases shown in 4. Results for different standard deviations and orders of directional filter are show in Fig. 5.

## 5 Discussion

The results that are shown in Fig. 5 depict better performance of system with directional filtering – values $a_1 < 1$ is related to the reduction of position errors according to formula (11). Low–valued noise distortions (standard deviation $< 0.3$) cases are better processed with the use of system without directional filtering. This is the result of line blurring introduced by filter. Larger noise is the main source of the position errors so filtering adds blurring to the line and background area, reducing errors. Larger noise gives about 20% reduction of the position errors in horizontal direction, which is related to $a_1 \approx 0.8$ value. Small minimum could be observed for filter order $3 - 7$ range and higher standard deviations (Fig. 5). Such filters orders are optimal and increasing order reduces performance due to line direction changes. Grayscale image could be used by the Viterbi algorithm so Euclidean metric is used. This is much better solution in comparison to the binary images processing. The Viterbi algorithm is one of the TBD (Track–Before–Detect) algorithms and allows the processing of signals hidden in noise. TBD processing assumes raw signal processing that has superior performance over conventional Detection and Tracking approach, that uses threshold algorithm. The obtained system is a kind of hierarchical TBD system [12]. Additional improvement of obtained results could be achieved using the estimated line filtering [14].

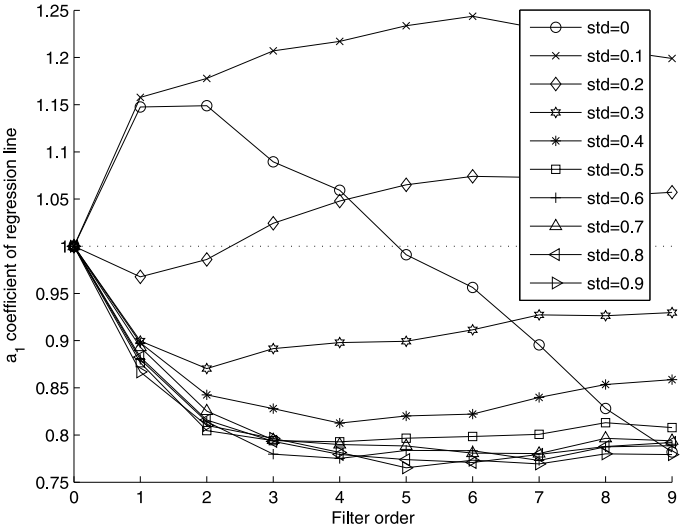


**Fig. 5.** Example results for Monte Carlo test for different standard deviations and orders of directional filter

## 6 Conclusions

Proposed directional filtering for the Viterbi algorithm input image preprocessing improves the estimation of lines. Such approach could be applied not only for the line following robots, but also in medical image processing for example [15]. Proposed approach extends possibilities of the application of the Viterbi based TBD algorithm by the application of the directional filtering that could be also considered as first TBD algorithm. The direction changes of the line is supported by the Markov model [16] processed by the Viterbi algorithm. Another preprocessing schemes could be applied for different types of lines [17,18].

The implementation of proposed algorithm does not requires high computation power and typical microcontrollers or DSP (Digital Signal Processors) could be applied for the real–time processing. Some DSP's have hardware unit or dedicated instructions for the acceleration of implementation of the Viterbi algorithm, that could be considered also. The direction filtering could be implemented using vertical instructions that are supported by all SIMD (Single–Instruction Multiple–Data) processors including VLIW (Very Long Instruction Word), additionally. Recursive processing of directional MA filers should be considered for the efficient implementation.

**Acknowledgment.** This work is supported by the UE EFRR ZPORR project Z/2.32/I/1.3.1/267/05 "Szczecin University of Technology – Research and Education Center of Modern Multimedia Technologies" (Poland).

## References


1. Horan, B., Najdovski, Z., Black, T., Nahavandi, S., Crothers, P.: Oztug mobile robot for manufacturing transportation. In: IEEE International Conference on Systems, Man and Cybernetics (SMC 2011), pp. 3554–3560 (2011)
2. Hasan, K., Nahid, A., Mamun, A.: Implementation of autonomous line follower robot. In: IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision, pp. 865–869 (2012)
3. Ishikawa, S., Kuwamoto, H., Ozawa, S.: Visual navigation of an autonomous vehicle using white line recognition. IEEE Transaction on Pattern Analysis and Machine Intelligent 10(5), 743–749 (1988)
4. Ollis, M.: Perception Algorithms for a Harvesting Robot. CMU-RI-TR-97-43, Carnegie Mellon University (1997)
5. 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)
6. Astrand, B., Baerveldt, A.: A vision-based row-following system for agricultural field machinery. Mechatronics 15(2), 251–269 (2005)
7. Zhang, J., Chambers, A., Maeta, S., Bergerman, M., Singh, S.: 3d perception for accurate row following: Methodology and results. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, November 3-7, pp. 5306–5313 (2013)
8. Okarma, K., Lech, P.: A fast image analysis for the line tracking robots. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 329–336. Springer, Heidelberg (2010)
9. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13(2), 260–269 (1967)
10. Stone, L., Barlow, C., Corwin, T.: Bayesian Multiple Target Tracking. Artech House (1999)
11. Mazurek, P.: Optimization of bayesian track-before-detect algorithms for GPGPUs implementations. Electrical Review R. 86(7), 187–189 (2010)
12. Mazurek, P.: Hierarchical track-before-detect algorithm for tracking of amplitude modulated signals. In: Choraś, R.S. (ed.) Image Processing and Communications Challenges 3. AISC, vol. 102, pp. 511–518. Springer, Heidelberg (2011)
13. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House (1999)
14. Frejlichowski, D.: Application of the curvature scale space descriptor to the problem of general shape analysis. Electrical Review 88(10 B), 209–212 (2012)
15. Scott, T.A., Nilanjan, R.: Biomedical Image Analysis: Tracking. Morgan & Claypool (2005)
16. Mazurek, P.: Code reordering using local random extraction and insertion (LREI) operator for GPGPU-based track-before-detect systems. Soft Computing 18(6), 1095–1106 (2013)
17. Mazurek, P.: Comparison of different measurement spaces for spatio-temporal recurrent track-before-detect algorithm. In: Choraś, R.S. (ed.) Image Processing and Communications Challenges 3. AISC, vol. 102, pp. 157–164. Springer, Heidelberg (2011)
18. Mazurek, P.: Preprocessing using maximal autocovariance for spatio-temporal track-before-detect algorithm. In: Choras, R.S. (ed.) Image Processing and Communications Challenges 5. AISC, vol. 233, pp. 45–54. Springer, Heidelberg (2014)