Tracklet-Based Viterbi Track-Before-Detect Algorithm for Line Following Robots

Grzegorz Matczak and Przemysław Mazurek

Abstract Line following robots could be applied in numerous applications with artificial or natural line. The proposed algorithm uses tracklets and Cartesian-to-polar conversion together with Viterbi algorithm for the estimation of line. The line could be low contrast or deteriorated and Monte Carlo tests are applied for the analysis of algorithm properties. Two algorithms are presented and compared—Viterbi and proposed Tracklet-based Viterbi Track-Before-Detect algorithms. Both of them are evaluated and properties are presented. The proposed algorithm could be better for smoother lines if higher noise disturb images.

1 Introduction

Line following robots are applied in numerous industrial [3] and entertainment applications. They use vision-based systems, that could be very simple or sophisticated. Simplest line following robots have applied two optical senors for the estimation of the position over the line. More advanced approach is based on the linear senor (1D camera) located under the robot. Cameras could be applied for the forward looking with perspective view of the line. This approach improves response of overall system, because the estimation of trajectory before the robot could be possible. Larger view allows better determination of line if distortions of many types occur. Some recent line following robots use 3D depth cameras for the estimation of the line. The main problem of line following robots is the vision system that should be robust for the poor light conditions and low line-to-background contrast. Such system should fulfill requirements of the real-time processing for constant speed maintaining. High-speed

G. Matczak · P. Mazurek (✉)
Department of Signal Processing and Multimedia Engineering,
West–Pomeranian University of Technology, Szczecin,
26. Kwietnia 10 Street, 71126 Szczecin, Poland
e-mail: przemyslaw.mazurek@zut.edu.pl

G. Matczak
e-mail: grzegorz.matczak@gmail.com

R. Burduk et al. (eds.), *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, Advances in Intelligent Systems and Computing 403, DOI 10.1007/978-3-319-26227-7_61

line following robots are in the area of interest of robotics enthusiasts and they are verified in numerous contests also. The importance of line following robots is not reduced by the availability of alternative navigation techniques. Satellite navigation (GPS) systems are not feasible for indoor applications. They have limited performance for outdoor scenarios and the real-time position is estimated with 2 m quality for single frequency receiver (L1) using standard (consumer grade) receivers. The position error could be reduced to 0.1 m for special civil receiver that are very costly. Moreover, limited availability of such receivers is related to political reasons. SLAM (Simultaneous Localization and Mapping) techniques are more costly due to higher computation power demands and line following robots are much simpler. This line could be artificially made or natural one.

1.1 Related Works

Line following robots are proposed in 60s of the previous century and the most notable is the Stanford Cart [14]. The area of applications related to harvesting and similar task, where the line is not specified directly, is considered in [13]. Road line estimation is necessary for automatic control of vehicle or is applied in lane departure warning system [15] using Hough Transform. Agricultural applications need natural lines estimation that could be low contrast [1] and the estimation of line is necessary even in LIDAR systems [17]. Recent works provide the approach based on the application of Track-Before-Detect algorithms [2] that improves lines estimation quality, so even lines hidden in the noise background could be detected [9, 10]. Additional trajectory filtering [4] may improve results independently on noise suppression in the spatial domain.

1.2 Content and Contribution of the Paper

The Viterbi algorithm is originally applied for time series with unknown multiple states at specific moments [16]. Replacing of such state by the 2D position is possible and the line estimation is possible for low curvatures. Directional filtering (FIR-based) that is parallel to the line direction improves the line estimation [9]. Local directional filtering is necessary with fixed direction for better estimation of curvatures. Such approach is important for the computation cost reduction also, so Viterbi algorithm is not applied for every image row. The idea of hierarchical processing of such data is introduced in [6]. Two filtering approaches are possible. First one is computed using original image and second one uses local Cartesian-to-polar transformation. The second approach is assumed in this paper and is considered in Sect. 2. Viterbi algorithm is applied for the selection of the most probable path and is considered in Sect. 3. Monte Carlo approach is applied for the numerical evaluation of algorithm performance and is considered in Sect. 4. The discussion is provided in Sect. 5 and the final conclusions are provided in Sect. 6.

2 Measurement Space Transformation

The acquired image frames are rectangular with Cartesian coordinates. Viterbi algorithm, discussed in next section, process them using small length track (tracklets). Every tracklet value is related to the average value of pixel related to the starting particular position and orientation:

$$d_{\{x,y,\phi\}} = \int_{l_\phi} I\left((x,y) \in l_\phi\right) \mathrm{d}l_\phi, \tag{1}$$

where l_ϕ is the line oriented with some angle direction ϕ and $\phi \in \langle -\alpha, \alpha \rangle$ Cartesian-to-polar transformation could be applied for limited angle range. The cone of analysis in such transformation is reduced to $\langle -\alpha, \alpha \rangle$. Multiple tracklets are processed starting from the particular pixel (x, y) with assumed cone (Fig. 1). This cone could be modified to triangle for the equalization of tracklets length in vertical direction. Example input image and the result of Cartesian-to-polar conversion with cone area are shown in Fig. 2. Such conversion changes sloped lines to straight lines that allows the application of Viterbi algorithm in more straightforward way like it is considered in [9, 10]. The cone (Fig. 2) in converted to the rectangular region corresponding with $\langle 180° - 60°, 180° + 60° \rangle$ range and this certain rectangular area is processed. Such reformulation simplifies the implementation of Viterbi algorithm, that is important in real-time processing applications. White areas present in Figs. 1 and 2 are omitted. Measurement space transformation could be applied locally with equal vertical length of tracklets what is computationally practicable.

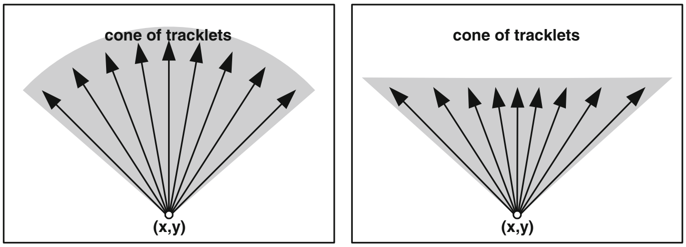


Fig. 1 Possible tracklets for particular pixel: equal length of tracklets (*left*) and equal vertical length of tracklets (*right*)

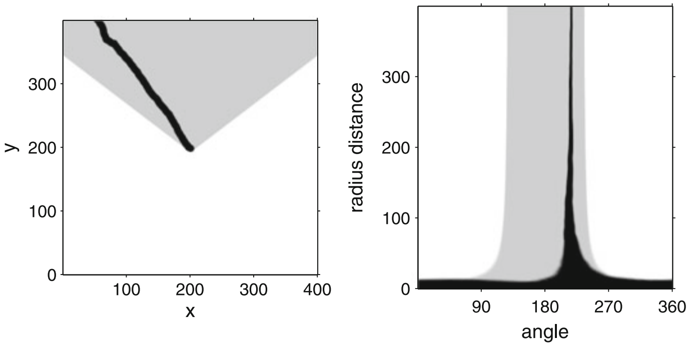
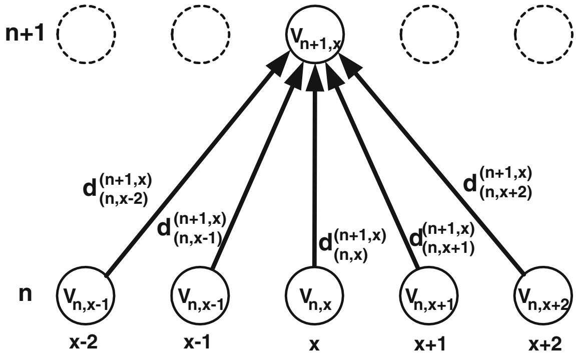


Fig. 2 Input image with line (*black*) and cone (*gray*) of analysis (*left*) and result of Cartesian-to-polar conversion (*right*)

3 Viterbi Algorithm

Viterbi algorithm could be applied for image processing using two approaches. First approach assumes direct relation between pixels and states. Second approach uses transformed image space using directional filtering for example, so tracklets could be processed. Viterbi algorithm uses two processing phases for particular starting row of pixels. The first phase uses forward processing and the second is backward for the best trajectory (best tracklets set) obtained from the first forward phase. Obtained result is the single tracklet related to the single starting pixel of the particular row. Such tracklets allows the robot motion control. Tracklets possible transitions are defined by the trellis. Example part of trellis is shown in Fig. 3 for two rows of pixels/nodes (n and $n+1$). There are n_{max} of rows in forward direction and Cartesian-to-polar conversion is well fitted to this definition of the Viterbi algorithm. The length of tracklets could be variable, but fixed is possible especially for polar measurement space. The transition cost is assumed as a value of particular tracklet which is added to the value of node. This node is related to the 2D position of tracklet beginning.

Fig. 3 Local paths in example trellis



Multiple tracklets are merged in nodes of the next row and only the maximal or minimal values are preserved. The minimal values are preserved in this paper for the following settings: grayscale image with values from the normalized $\langle 0 - 1 \rangle$ range; white color (value 1) corresponds to the background; the black color corresponds to the line (value 0), for the ideal case. The local cost for modified cone (triangle) is corrected due to different length of tracklets between nodes of two following rows:

$$d_{corrected\{.,\phi\}} = \frac{d_{\{.,\phi\}}}{|d_{\{.,\phi\}}|}. \tag{2}$$

The image is processed using trellis and the example part of the trellis is show in Fig. 3. The Viterbi algorithm process n_{max} rows of nodes starting from the bottom row ($n = 1$) and the obtained result is the transition between first and second row only. Further transitions between rows $n = 2 \cdots n_{max}$ are processed for the selection of this particular transition and are not reused in next steps (e.g., $n = 2 \cdots n_{max} + 1$). Overall process is repeated for next sets (e.g., $n = 2 \cdots n_{max} + 1$) without relation to the previous results. Initialization is the assignment of zero value to the total cost V for nodes of the first row ($n = 1$):

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

The local cost d (exactly $d_{corrected}$) is added to previous nodes:

$$V_{n+1,x} = \min\left(V_{n,x+g} + X_{n,x+g}\right), \tag{4}$$

but the minimal cumulative cost is preserved only. The transition is preserved in L using the following formula:

$$L_n^{n+1,x} = \arg\min_g \left(V_{n,x+g} + X_{n,x+g}\right). \tag{5}$$

The first phase of the Viterbi algorithm is computed up to n_{max}. The solution for the first phase (path) is the node with minimal value:

$$P_{opt} = \max\left(V_{n=n_{min},.}\right), \tag{6}$$

that is related to the x–position:

$$x_{n=n_{max}} = \arg\min_x \left(V_{n=n_{max},x}\right). \tag{7}$$

The Selection of the final node allows backward processing for the selection of transition between two first rows. The following recursive processing formula:

$$x_{n-1} = x_m + L_{n-1}^{n,x}, \tag{8}$$

for successively decremented row numbers:

$$n = n_{max}, \ldots, 2. \tag{9}$$

Two results are obtained—most probable node and transition to the next row.

4 Monte Carlo Test of Tracklets-Based Algorithm

The Tracklet-based algorithm reduces number of the processed row by the Viterbi algorithm (Fig. 4) and introduces directional filtering by the accumulation of values related to the path between rows. The increased number of rows n is responsible for better noise suppression. The filtering expect low probability of line direction changes and such example line is show in Fig. 4. Monte Carlo approach allows the estimation of the properties of algorithms because is unbiased. The comparison of algorithms is possible also for identical input images. The generator of images allows the testing for different controllable conditions, which is very important for the algorithm evaluation. The test scenario is based on the analysis of image that has 200 × 60 pixel resolution. An example results of the single case are depicted in the Fig. 5. The line has standard deviation 1.0 and image is disturbed by a few lines with standard deviation selected by additive Gaussian noise. Two algorithms are compared: Viterbi and Tracklet-based tracking for exactly the same input images. There are 1400 testing scenarios for specified Gaussian noise (std. dev. 0–0.8), with two different line direction change probabilities {0.25, 1.25}. The main test was related to different number of rows processed by the Tracklet-based algorithm. Comparative results for the Viterbi algorithm are provided also (Fig. 8). The cumulative computation time

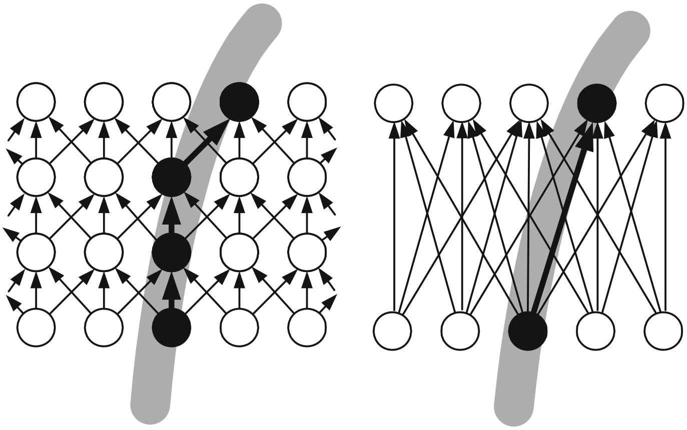


Fig. 4 Processed path in the Viterbi Algorithm (*left*) and in the tracklet-based Viterbi algorithm (*right*)

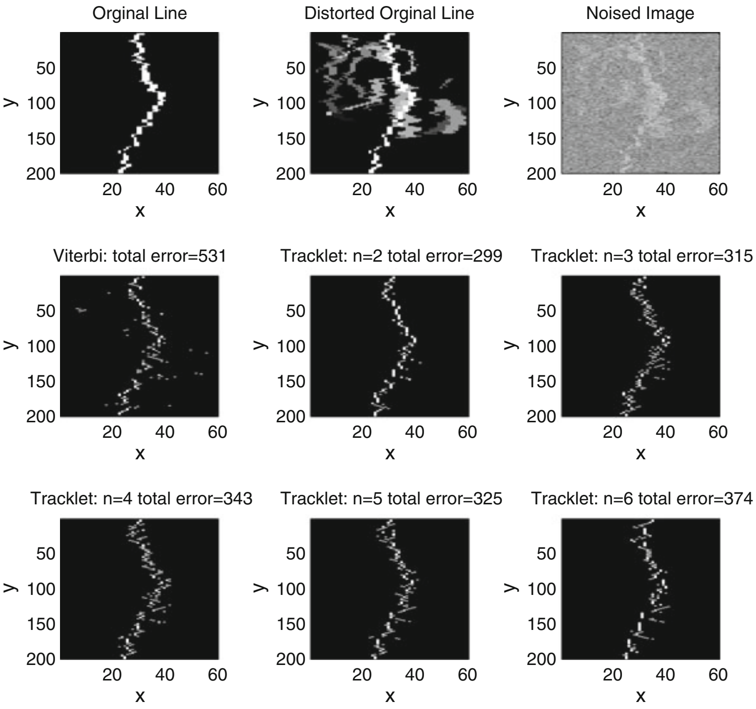


Fig. 5 Example results for specific image—std. dev. 0.45 and line direction change probability 0.25

is 500 h of 2.4 GHz core using Matlab for this test, but the real-time processing is possible for robot using optimized code. The regression line is shown in Fig. 6 for comparison both algorithms. The slope and position of this line shows the advantages of the Tracklet-based algorithm, because the mean position error for horizontal direction is smaller.

5 Discussion

Very interesting results are shown in Fig. 7. Two histograms are depicted for both algorithms and they are different. Better distribution of errors is achieved for the Tracklet-based algorithm because probability of errors is exponential (smaller errors are more probable). The errors for the Viterbi algorithm have mean value near to 5.

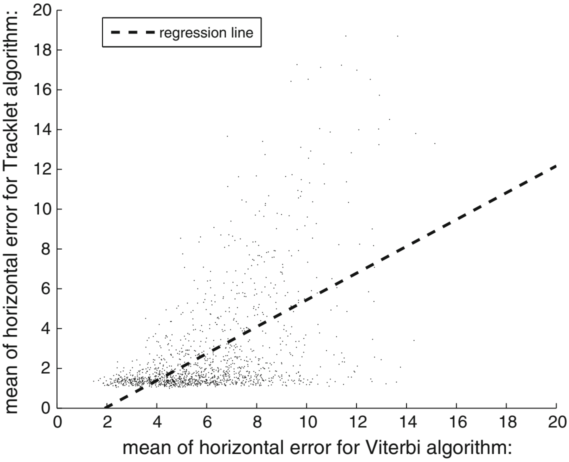


Fig. 6 Comparison of mean errors for Viterbi and tracklet-based algorithms (Monte Carlo test for std. dev. 0.45 and line direction change probability 0.25)

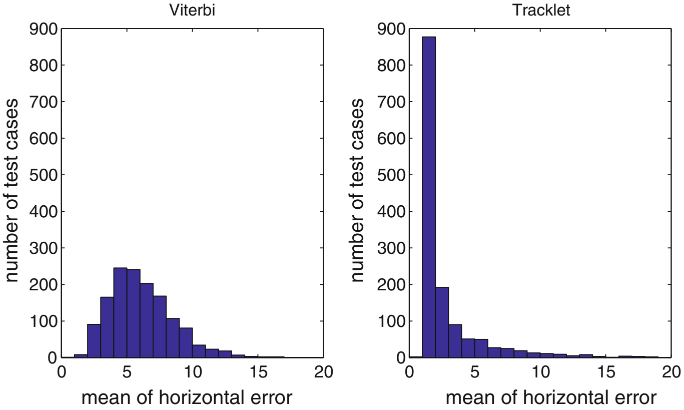


Fig. 7 Histogram of mean error for Viterbi and tracklet-based algorithm (Monte Carlo test for std. dev. 0.45 and line direction change probability 0.25)

Final test (Fig. 8) shows better performance of the Viterbi algorithm for very small noise of images. The dynamic of the line influences the results. Small *n* could be selected for higher probabilities of line direction changes. Longer integration (larger *n*) is recommended for the smooth lines.

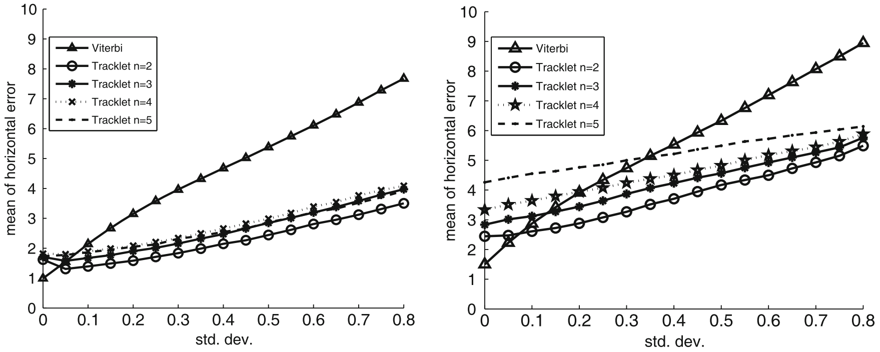


Fig. 8 Mean error for Viterbi and tracklet-based algorithms (Monte Carlo test for variable noise and line direction change probability 0.25 (*left*) and 1.25 (*right*))

6 Conclusions

Line following robots could be improved using vision systems based on 2D camera. Difficult light conditions or line deteriorations are not significant problem for advanced image processing algorithms. Track-Before-Detect (TBD) approach allows processing of weak lines, even below the noise floor. There are applications where only noise is observed, so object's signal is the noise pattern [8, 11]. Such case needs preprocessing of measurements independently on impulse noise suppression. Textures suppression algorithms [12] are also important for the reduction of spatial interferences. The cost of computation for TBD algorithms is significant, but modern available computing devices like SIMD-based CPU cores, GPGPUs and FPGA devices support real-time processing. Optimization techniques for TBD algorithms implementations are also possible [5, 7]. Viterbi algorithms could be accelerated using dedicated hardware that is very often available in modern DSPs. This algorithm is used for the digital communication applications, and there are dedicated instructions or coprocessors in DSPs. Monte Carlo approach allows testing of Viterbi and proposed Tracklet-based Viterbi algorithms and fitting parameters for the best performance and computational cost reduction. This approach allows the comparison of algorithms and the selection of better if variable conditions related to estimated background noise occurs. Such technique of image analysis allows the design better line estimation. Further works will be related to the implementation of mentioned algorithms for the line following robot.

Acknowledgments 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. Astrand, B., Baerveldt, A.: A vision-based row-following system for agricultural field machinery. Mechatronics **15**(2), 251–269 (2005)
2. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House, Norwood (1999)
3. Horan, B., Najdovski, Z., Black, T., Nahavandi, S., Crothers, P.: Oztug mobile robot for manufacturing transportation. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2011). pp. 3554–3560 (2011)
4. Marchewka, A.: Crack detection on asphalt surface image using local minimum analysis. Adv. Intell. 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.: Hierarchical track-before-detect algorithm for tracking of amplitude modulated signals. Adv. Intell. Soft Comput. **102**, 511–518 (2011)
7. 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)
8. Mazurek, P.: Track-before-detect filter banks for noise object tracking. Int. J. Electron. Telecommun. **59**(4), 325–330 (2013)
9. Mazurek, P.: Directional filter and the viterbi algorithm for line following robots. In: Chmielewski, L., Kozera, R., Shin, B.S., Wojciechowski, K. (eds.) Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 8671, pp. 428–435. Springer, Berlin (2014)
10. Mazurek, P.: Line estimation using the viterbi algorithm and track–before–detect approach for line following mobile robots. In: Proceedings of the 19th International Conference on Methods and Models in Automation and Robotics. pp. 788–793 (2014)
11. Mazurek, P.: Preprocessing using maximal autocovariance for spatio-temporal track-before-detect algorithm. Adv. Intell. Syst. Comput. **233**, 45–54 (2014)
12. Okarma, K., Frejlichowski, D., Czapiewski, P., Forczmański, P., Hofman, R.: Similarity estimation of textile materials based on image quality assessment methods. Lecture Notes in Computer Science **8671**, 478–485 (2014)
13. Ollis, M.: Perception Algorithms for a Harvesting Robot. CMU-RI-TR-97-43. Carnegie Mellon University, Pittsburgh (1997)
14. Schmidt, R.J.: A Study of the Real-time Control of a Computer-Driven Vehicle, Ph.D. thesis. Stanford University, Stanford (1971)
15. Taubel, G., Yang, J.S.: A lane departure warning system based on the integration of the optical flow and Hough transform methods. In: Proceedings of the 2013 10th IEEE International Conference on Control and Automation (ICCA) Hangzhou, China, June 12–14, 2013. pp. 1352–1357 (2013)
16. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory **13**(2), 260–269 (1967)
17. Zhang, J., Chambers, A., Maeta, S., Bergerman, M., Singh, S.: 3D perception for accurate row following: Methodology and results. In: Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) November 3-7, 2013. Tokyo, Japan. pp. 5306–5313 (2013)