The Effective Use of the Exhaustive Search Block Matching Algorithm in Railway Line Tracking

Enock T. Chekure
Electrical Engineering Department
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
Email: enoctaffie@gmail.com

Kevin A. Naudé
Computing Sciences Department
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
Email: kevin.naude@gmail.com

Peter Freere
Electrical Engineering Department
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
Email: pfreere@gmail.com

***Abstract*—This paper gives the results of an investigation on the use of exhaustive block matching techniques in detecting railway-line tracks in a video sequence with the view of implementing the outcome in a real-time system. The main focus of the research is to implement the exhaustive search block matching algorithm optimally and resolve any implementation problems that may arise. The algorithm is implemented on a video sequence that has undergone inverse perspective mapping. The second highlight of this research is to eliminate many video processing techniques, this is done in order to compensate for the computational expense for which the exhaustive search block matching algorithm is known for. The procedure entails traversing a very small block from the previous frame on the top and bottom ends of the current frame and marking the coordinates where the cost function is the minimum.**

Keywords: Block Matching Algorithm(BMA), Inverse perspective mapping, lane tracking, Mean Absolute Difference(MAD), Sum of Absolute Difference(SAD)

I. Introduction

Block matching is a computer vision technique that is important in motion estimation. The best possible block matching can be found through exhaustive search [1], [2], [3], although an exhaustive search carries significant computational expense. It has been asserted that exhaustive search can be implemented in real time with the aid of specialised hardware [4].

The motive of this research is to detect obstacles on railway line tracks. The detection of obstacles is a secondary process after railway line tracks have been properly tracked. The detection of obstacles will therefore be focused on the tracked railway line tracks with the hope of enhancing the processing time. This paper however elaborates on the implementation strategies aimed at reducing the the computational expense an exhaustive search block matching algorithm carries. Many researchers have argued that it is difficult to use exhaustive search in real-time processing [2], [3], [5] but they all agreed that the algorithm needs to be tweaked in order to perform in real time processing.

The size of the blocks that are being matched determine the computational cost of the algorithm [2], [6] . Hassen et al [2] derived an equation that gives the number of computations the algorithm performs which is shown in equation 1.

$$N = (2p + 1)^2 w^2 \tag{1}$$

Where:
p is the size of the search area.
w is the size of the matching block.
Both p and w are quantities in pixels in a 2D grid.

Several algorithms have been developed to reduce the block sizes and improve the searching criteria, among them there is:

- Three Step Search (TSS) [7].
- New Three Step Search (NTSS) [8].
- Simple and Efficient Search (SES) [9].
- Four Step Search (4SS) [10].
- Diamond search [11]
- Adaptive Rood Pattern Search (ARPS) [12]

Since research has established that exhaustive search produces the best task performance, that is, finding the best match of all block matching algorithms [1], [2], [3], [13], this was the basis of its selection in this research work and to investigate on strategies that aim to reduce the computational cost to meet real time processing capability. Two approaches with significantly different sizes of matching blocks have been experimented with as illustrated in the following section. There has been a lot of research in lane detection and tracking though much of such research has been done in the context of road systems [14]. Some of the developments and concepts of such research have been applied here to the railway systems.

II. Methodology

A. Preprocessing

The preprocessing stage demarcates the region of interest (ROI) which is the region that is further processed in the successive image frames; figure 1 shows such a demarcated region of interest.

Inverse perspective mapping is used to obtain a *bird's eye* view of the ROI, as presented in figure 2. The assumption required do the inverse perspective mapping is that the surface within the ROI is a plane [15]. This assumption is reasonable in the sense that gradient changes in the vicinity of train tracks is very slight. The focal length of the camera, the height of the camera above the ground level and the camera tilt angle provides enough information to produce the two coordinates of the inverse perspective mapped point of each pixel in the ROI. Equation 2 gives the x-coordinate, x^* and equation 3

978-1-5386-2775-4/17/$31.00 ©2017 IEEE

Fig. 1: ROI demarcation

Fig. 2: Bird's eye view of ROI

gives the y-coordinate, y^* of the inverse perspective mapped point of each pixel [15].

$$x^* = H\frac{xsin\theta + fcos\theta}{-ycos\theta + fsin\theta} + d \tag{2}$$

$$y^* = H\frac{ysin\theta + fcos\theta}{-ycos\theta + fsin\theta} + d \tag{3}$$

Where:
H is the height of the camera above the ground,
x is the x-coordinate of each pixel in the ROI,
y is the y-coordinate of each pixel in the ROI,
f is the focal length of the camera,
d is a constant that ensures that x^* and y^* are not less than zero and is given by equation 4.

$$d = |H\frac{sin\theta + fcos\theta}{fsin\theta - cos\theta}| + 1 \tag{4}$$

B. Exhaustive Search BMA

This research took two approaches for selecting the matching blocks which were experimented with. The first approach used a single matching block in a single run and the second approach involved doing two runs, one at the top and one at the bottom of the search area, see an illustration in figure 7. The algorithms for doing the exhaustive search in each search area for the respective approaches are given in algorithm 1 and algorithm 2.

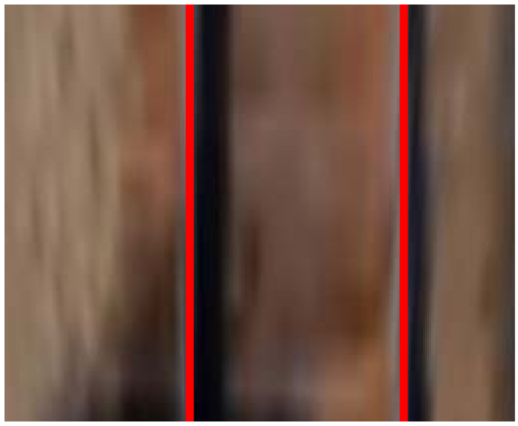


Fig. 3: Good tracking of the track

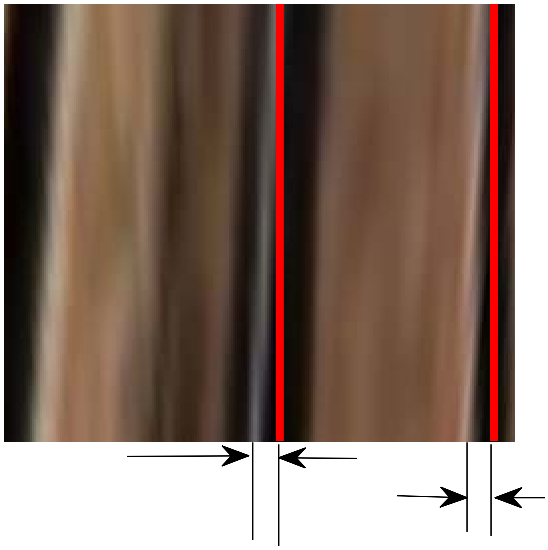


Fig. 4: Imperfect tracking on a curve of the track

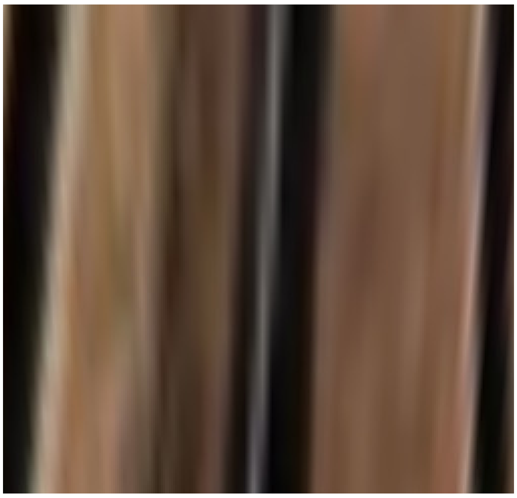
$$SAD = \sum_{x=1}^{M}\sum_{y=1}^{N} |I_n(x+k, y+k) - (I_{n-1}(x,y))| \tag{5}$$

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N} (I_n(x+k, y+k) - (I_{n-1}(x,y)))^2 \tag{6}$$

$$MAD = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N} |I_n(x+k, y+k) - (I_{n-1}(x,y))| \tag{7}$$

Where:
$I_n(x+k, y+k)$ is the current frame.
$I_{n-1}(x,y)$ is the previous frame.

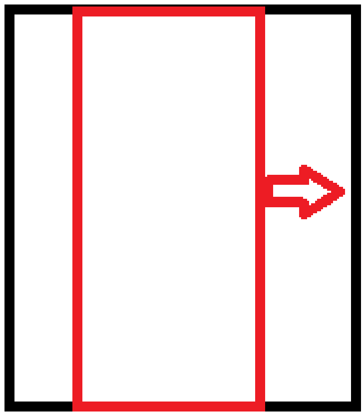
(a) Railway line track orientation on a straight stretch



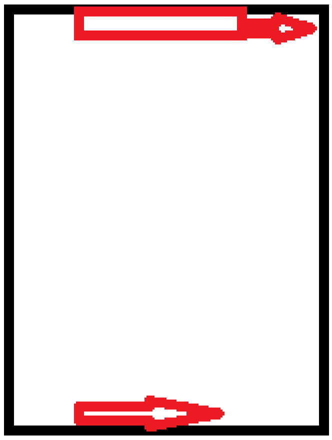
(b) Railway line track orientation when negotiating a curve

Fig. 5: Railway line track orientation changes

Fig. 6: Good tracking on a curve of the track



(a) Algorithm 1



(b) Algorithm 2

Fig. 7: Pictorial representation of the sizes of the blocks and how they traversed the current frame

Fig. 8: Example of a curved stretch of the track

M is the height of the matching block.
N is the width of the matching block.
The two approaches are outlined in the following subsections.

1) Full ROI Height Pixel Block: In this approach the matching block height equals the search area height which is 255 pixels. The width of the matching block is selected to correspond to the outside width of the railway line track, which is 110 pixels. The matching block is traversed across the search area from the left edge to the right edge. The matching block is moved one pixel at a time. The matching block is obtained from the previous search area. An illustration of the matching block and the search area are provided in figure 7a where the matching block is illustrated as a red box.

The reason for selecting this approach is to provide the reference point as it attempts to use as much information as is available in the ROI to track motion between two search areas. The full ROI height pixel block approach uses the entire ROI as the search area which according to Yaakob et al [13] enhances the chances of obtaining a valid match at the expense of the process performance, that is, the computational cost.

As the matching block is traversed, the cost function is calculated, initially the least cost function is equal to the cost function of the first calculation. As the matching block is traversed and more cost function calculations are made, if the cost function is less than the least cost function the value of the least cost function is updated and the top coordinates of the current block are recorded. The final coordinates, where the cost function is the least are used as the coordinates of the matching block when the following search area is being processed; this is how the tracking is done in this algorithm.

2) Narrow Top and Bottom Blocks: This approach entails using two very small matching blocks, with height restricted to 10 pixels, and the width corresponding to the outside width of the railway line tracks. The narrow top and bottom blocks approach involves traversing the matching block once for the top 10 pixels of the search area and once for the bottom 10 pixels of the search area, so as to track horizontal deviation at different distances ahead of the locomotive. The illustration of the matching block and the search area is as in figure 7b. The matching block is illustrated by the red box and the arrow at the bottom illustrates traversing the matching block the bottom 10 pixels. The procedure for the tracking is exactly the same as for the full ROI height pixel block approach. The only difference is four coordinates are recorded in each current search area, two when the top 10 pixels are traversed and two when the bottom 10 pixels are traversed.

The narrow top and bottom blocks approach seeks to establish the difference in the number of calculations when the block was only made ten pixels high. This is good enough a height to establish similarities between the matching block and the current search area. There is no need to traverse the matching block more than these two times since the railway line tracks run from the top to the bottom of each frame. The reduction of the matching block size and traversing the matching block twice means a reduction in the number of calculations.

3) Cost Function: The cost function represents the similarity between the matching block and the search area, there are three formulae that are normally used to calculate this cost function. The mean squared error (MSE), the mean absolute difference (MAD) and the sum of absolute difference (SAD). Their equations are shown in equation 6, equation 7 and equation 5 respectively. The square operation on the differences introduces a computational complexity in calculating the cost function using MSE [16] therefore Wong et al recommended the use of MAD in calculating the cost function. In line with the requirement to keep the number of calculations and their complexity at the minimum [5], [16], SAD was used in this research work to compute the cost function as both MSE and MAD introduce a complexity in the calculations with the division operation [5].

Algorithm 1 ES BMA Block height equal to the entire frame height

```
read current frame
read the previous frame
set a very high leastSAD value
if it's the first iteration then
    set the block according to predetermined position
else
    set the block according to the motion vector
for the width of the current frame do
    traverse the block on the current frame
    calculate the SAD
    if SAD is smaller than leastSAD value then
        update leastSAD
        mark the columnns
Calculate the bottom columns and plot straight lines
```

Algorithm 2 ES BMA Block hieght equal to 10 pixels

```
read current frame
read the previous frame
set a very high leastSAD value
if it's the first iteration then
    set the block according to predetermined position
else
    set the block according to the motion vector
top of current frame:
for the width of the current frame do
    traverse the block on the current frame
    calculate the SAD
    if SAD is smaller than leastSAD value then
        update leastSAD
        mark the columnns
bottom of current frame:
for the width of the current frame do
    traverse the block on the current frame
    calculate the SAD
    if SAD is smaller than leastSAD value then
        update leastSAD
        mark the columnns
Plot straight lines from the marked column points
```

C. Results and Discussions

The full ROI height pixel block approach gives good tracking results as it can be established from figure 3 but only on straight stretches of railway line. This approach does not produce good tracking results on curved stretches, even though the ROI is itself of narrow height, as can be seen from figure 4 where a small error between the railway line and the tracked line emerged. This is as a result of only top coordinates of where the the least cost function occurs being used to plot where the railway line tracks are. The changes in the orientation of the railway line tracks as a curve is negotiated by the locomotive introduces the error at the bottom of the frame as the plotting procedure will assume the lines are perfect vertical lines. Figure 5 gives an illustration of the changes in the orientation of the railway line tracks. Ideally the inverse perspective mapping should give straight lines of the ROI, but practically the lines always change orientation as the locomotive negotiates a curve. The illustration of a curved stretch is shown in figure 8.

The imperfection in tracking shown in figure 4 is however rectified by using the narrow top and bottom blocks approach. The fact that the matching block is traversed across the frame twice, once at the top and once at the bottom of the search area enables the operation to get the top and bottom coordinates of the railway line track in each ROI. This therefore addresses the problem of railway line tracks changing the orientation on curved stretches which resulted in the error in tracking. With the four coordinates of the railway line recorded, the plotting process followed perfectly where railway line tracks are to give a good tracking as shown in figure 6.

The algorithms in this research work were implemented in MATLAB on a general purpose computer with 8GB RAM, i5 5200U CPU running at 2.2GHz and running Windows 7™; this computer was running other functions in the background and algorithm 1 took 0.691s to run each frame on average and algorithm 2 took on average 0.648s which can lead to a conclusion that, with further optimisation and good hardware it should be possible to run the two algorithms in real time. The full height ROI pixel block approach did 1.33×10^{19} calculations on each frame that was reduced significantly to 6.3×10^{13} calculations by the narrow top and bottom blocks approach which resulted in the processing time per frame being reduced by 0.05s.

III. Conclusion and Future Work

The inverse perspective mapping makes the implementation of the tracking of the railway line tracks simple in the two approaches used in this research. Searching for railway lines allows a simplified exhaustive search process in contrast to, for example searching for radiata pine trees. The tracking process entails obtaining the coordinates where the best match occurred and then simply plotting vertical lines. The results show that the use of the narrow top and bottom blocks approach provides stronger task performance. This approach is also likely to enable further optimisation. Future work will focus on optimisation. The approach to be followed involves doing the exhaustive search from the left edge of the current frame only once in the first search area. Once a best match has been obtained in the next frame the exhaustive search should start somewhere close to the coordinates where the best match occurred.

This research has proved that it is possible to use an implementation strategy for reducing the computational cost of the exhaustive search block matching algorithm but reliance on good hardware to achieve real time processing is the option [4]. The good tracking of the railway line tracks using the narrow top and bottom blocks approach is expected to serve as an enabler in future work of detecting obstacles on the railway line tracks. The railway line tracks can be tracked very well therefore the scope of view of where the obstacles can be searched for can be reduced significantly.

References

[1] Barjatya, A: "Block matching algorithms for motion estimation", *Article In: IEEE Transactions Evolution Computation,* IEEE, 8(3), pp.225-239, 2004.

[2] Hassen, W and Amiri, H: "Block Matching Algorithms for motion estimation", *Proceedings of the: 7th e-Learning in Industrial Electronics (ICELIE) 2013, IEEE International Conference on,* (pp. 136-139) IEEE, November 2013.

[3] Khude, P.S and Pawar, S.S: "Object detection, tracking and counting using enhanced BMA on static background videos", *Proceedings of: In Computational Intelligence and Computing Research (ICCIC) 2013, IEEE International Conference on,* (pp. 1-4) IEEE 2013.

[4] Bartolini, F, Cappellini, V and Giani, C: "Motion estimation and tracking for urban traffic monitoring", *Proceedings of: Image Processing, 1996, International Conference on* (Vol. 3, pp. 787-790), IEEE 1996.

[5] Ni, T, Li, Q, Sun, L and Huang, L: "Dynamic Obstacle Detection Based on Background Compensation in Robots Movement Space", *Article In: MATEC Web of Conferences,* (Vol. 95, p. 08013, EDP Sciences 2017.

[6] Di Stefano, L and Viarani, E: "Vehicle detection and tracking using the block matching algorithm", *Proceedings of the: 3rd IMACS 1999,* IEEE, 1, pp.4491-4496 1999.

[7] Haskell, B.G, Puri, A. and Netravali, A.N: "Digital video: an introduction to MPEG-2", *Book,* Springer Science & Business Media, 1996.

[8] Li, R, Zeng, B and Liou, M.L: "A new three-step search algorithm for block motion estimation", *Article In: IEEE transactions on circuits and systems for video technology,* 4(4), pp.438-442, 1994.

[9] Lu, J. and Liou, M.L: "A simple and efficient search algorithm for block-matching motion estimation", *Article In: IEEE Transactions on Circuits and Systems for Video Technology,* 7(2), pp.429-433, 1997.

[10] Po, L.M and Ma, W.C: "A novel four-step search algorithm for fast block motion estimation", *Article In: IEEE transactions on circuits and systems for video technology,* 6(3), pp.313-317, 1996.

[11] Zhu, S and Ma, K: "A new diamond search algorithm for fast block matching motion estimation", *Article In: IEEE Trans. Image Processing,* vol. 9, No. 2, pp. 287-290, February 2000.

[12] Nie, Y and Ma, K: "Adaptive rood pattern search for fast block matching motion estimation", *Article In: IEEE Trans. Image Processing,* vol. 11, No. 12, pp. 1442-1448, December 2002.

[13] Yaakob, R, Aryanfar, A, Halin, A.A and Sulaiman, N: "A comparison of different block matching algorithms for motion estimation", *Article in: Procedia Technology,* Vol. 11, pp.199-205, 2013.

[14] Kaleli, F and Akgul, Y.S: "Vision-based RailRoad Track Extraction Using Dynamic Programming", *Proceedings of:12th Intelligent Transportation Systems ITSC'09, International IEEE Conference on,* IEEE, pp.1-6, 2009.

[15] Lin, C.C and Wang, M.S: "A vision based top-view transformation model for a vehicle parking assistant", *Sensors*, Vol. 12(4), pp.4431-4446.

[16] Wong, S, Vassiliadis, S and Cotofana, S: "A sum of absolute differences implementation in FPGA hardware", *Proceedings of:* 28th *Euromicro Conference* IEEE, pp. 183-188, 2002.