Vehicle Speed Estimation Using Optical Flow

Fábio Crestani, Daniel Rodrigues Pipa, and Ricardo Carnieri.

Abstract—**This article presents a digital image processing algorithm to estimate a vehicle speed using Lucas-Kanade Optical Flow. It was calibrated using images with known car speeds. The optical flow algorithm was improved with some statistical operations to refine the results for this specific case. The simulated results showed an acceptable average error.**

Keywords—Vehicle speed estimation, Lucas Kanade, Optical Flow, Median Estimator.

I. Introduction

The key to determine a speed based on image processing is the motion estimation of an image sequence. Some works, as Mao & Song [1] and Dailey & Li [2] have already estimated speeds using blob recognition for tracking the vehicles and applying geometrical operations with known camera parameters. The Doğan *et al.*'s [3] approach uses the Lucas-Kanade optical flow algorithm, though limited to a side view of the road.

We chose the Pyramidal Lucas-Kanade tracker algorithm. This method generates a dense vector field of the displacements for every pair of images. As noted by Oge [4], this algorithm is known as a hierarchical algorithm because it is first applied at a down-sampled version of an image frame and then refined using successively finer resolutions.

Our software uses this algorithm to create an initial field that is operated to provide a unique vector that estimates the speed of a vehicle in a sequence of frames. We assume there is only one vehicle visible in the image at a time.

II. Vector Selection

The first step of our software is to apply a median operation in the whole displacements vector, generating a unique vector that serves as a reference for further refine. From this, we tried two approaches to improve the result, as follow:

A. Applying a weighted median based on the angle with the reference vector;

B. Applying a simpler criterion, removing values whose angle is very different form the reference.

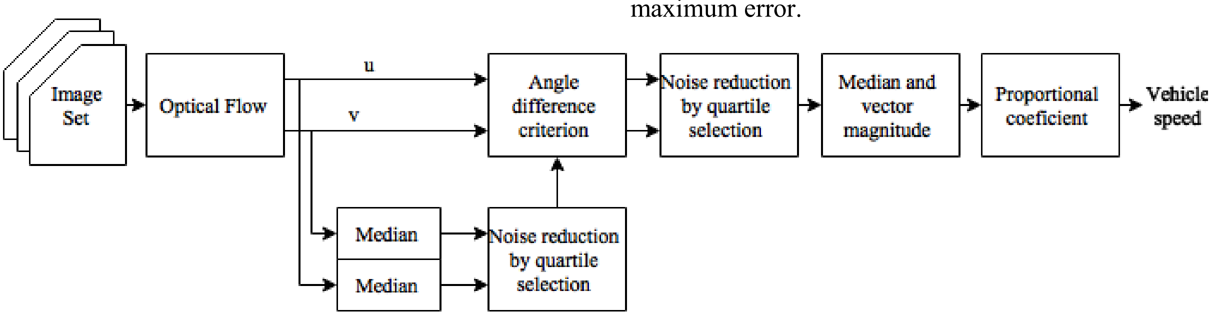

Fig. 2. System block diagram

A. Smooth weighting

The first approach consists in obtaining an error factor based on the angle between every vector i and the reference vector. From this value, it is calculated a weight w:

$$w_i = \cos(\theta_r - \theta_i) \tag{1}$$

where w is the weight, θ_r determines the expected direction of travel and θ_i is the angle of the i-th optical flow vector.

These values are computed as weights of a weighted median as proposed by Li & Osher in [5]. The Fig. 1 a. shows the resulting vectors of a real set of images and the Fig. 1 b. shows a representation of theses vectors and angles.

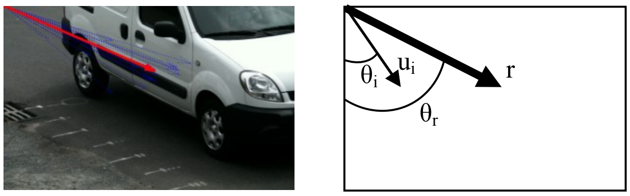

Fig. 1. a. Calculated reference vector b. Angle difference representation

The problem of this approach is that for every pixel of the frame is associated a vector and a weight. As the weighted median function iterates on the whole image, the processing time is quite long.

Trying to contour this, we applied the weighted median on a vector field generated by a down-sampled image frame. The computation time was significantly reduced, however the results were unsatisfactory. We observed that the down-sampled field lost much information about the motion of the object.

B. Hard selection

The second approach uses the same principle of a reference vector, however, it applies the following rule to every displacement vector:

$$w_i = \begin{cases} 0, & |\theta_r - \theta_i| > \varepsilon \\ 1, & otherwise \end{cases} \tag{2}$$

In other words, this rule sets to zero a vector whose angle difference with the reference is greater than a constant maximum error.

III. Noise Reduction

Analyzing the histograms of these vectors, we tried to extract the most relevant vectors based on its magnitude. The Fig. 3 shows the histograms of some frames.

We observed that a sequence of images without a car produces just noise with low magnitude vectors. Therefore, we decided to extract the relevant vectors getting the superior quantile of the distribution.

The last step was applying a median to these vectors, resulting in a unique vector that represents the estimated vehicle speed. The Fig. 2 presents a block diagram of the system using both the hard selection and noise reduction techniques.

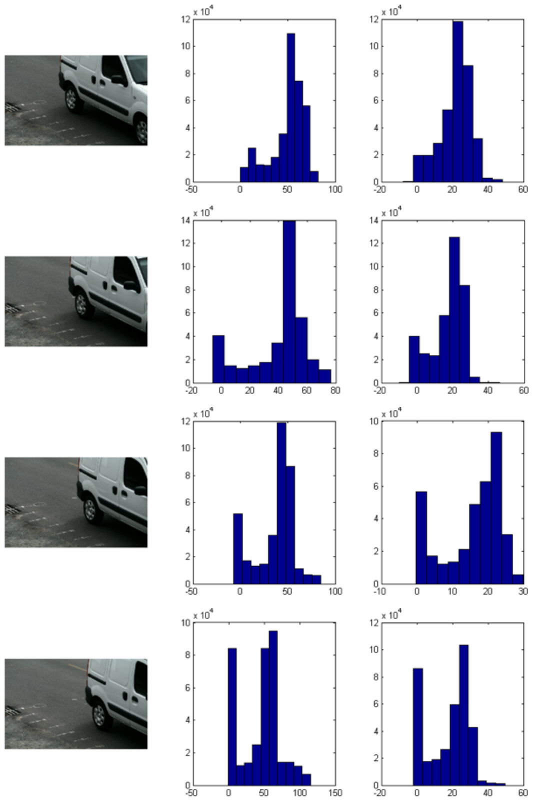

Fig. 3. Examples of image frames (left) and respective histograms of horizontal (center) and vertical (right) displacements vectors for 20 km/h.

IV. Simulation Results

In order to calibrate the system, we applied it on a small set of images with known vehicle speeds. Each one of these sets resulted on a coefficient from which a linear curve that relates the displacement vector with a real speed was fitted.

After, the algorithm was applied in some sequences with known speeds in order to determine an error factor.

Fig. 4 shows the results from the approach A and the Fig. 5 shows the results from the approach B.

Fábio Crestani and Daniel Rodrigues Pipa are with Departament of Electronic Engineering, Federal University of Technology – Paraná-PR, Brazil, E-mails: fabiocrestani@alunos.utfpr.edu.br, danielpipa@utpr.edu.br. Ricardo Carnieri is with, Gaussian Inteligência Computacional, Curitiba-PR, Brazil.

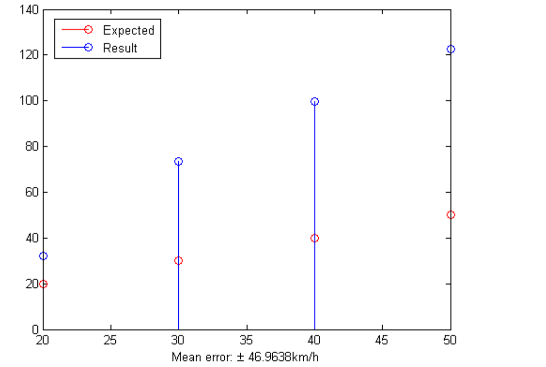

Fig. 4. Speed expected (red) and result of the smooth weighting (blue).

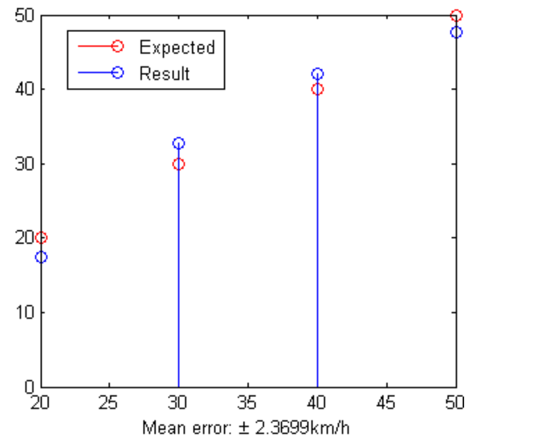

Fig. 5. Speed expected (red) and result of the hard selection (blue).

V. Conclusions

Although we used a small set of images for calibration, the algorithm showed a reasonable result in an interval between 20 km/h and 50 km/h. In the simulated test with real images, the result showed a mean error of 2.37 km/h, that is coherent compared with the results of [1] and [2], that showed, respectively, a mean error of 2.91 km/h and 1.12 km/h. A larger set of calibration images would improve the relation between the pixel displacement and the physical speed.

Other measure to enhance the robustness of the system would be an adaptive selection of the quantile based on some metrics of the histograms.

In future works we intend to develop a system with self-calibration which could auto adjust for some different camera positions.

References

[1] H. Mao, C. Ye, M. Song, J. Bu, and N. Li, *Viewpoint independent vehicle speed estimation from uncalibrated traffic surveillance cameras*” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, no. October, pp. 4920–4925, 2009.

[2] D. J. Dailey and L. Li, *An Algorithm to Estimate Vehicle Speed Using Un-Calibrated Cameras*, vol. 1, no. 2, pp. 441–446, 1999.

[3] S. Doğan, M. S. Temiz, and S. Külür, *Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera, Sensors*, vol. 10, no. 5, pp. 4805–4824, 2010.

[4] Oge, M. *Practical Image and Video Processing Using MATLAB*. Wiley-IEEE Press.2011.

[5] Li, Y. and Osher, S. *A New Median Formula with Applications to PDE Based Denoising*, Commun. Math. Sci., vol. 7, no. 3, pp. 741–753.