Vehicle Speed Detection from Camera Stream Using Image Processing Methods

Jozef Gerát, Dominik Sopiak, Miloš Oravec, Jarmila Pavlovičová
Faculty of Electrical Engineering and Information Technology/Slovak University of Technology in Bratislava,
Ilkovičova 3, 812 19, Bratislava, Slovak Republic
{xgerat; dominik.sopiak; milos.oravec; jarmila.pavlovicova}@stuba.sk

***Abstract*—The paper deals with the topic of detection of vehicle speed based on information from video record. In theoretical part we describe the most important methods, namely Gaussian mixture models, DBSCAN, Kalman filter, Optical flow. The implementation part is comprised of the architectural design and the description of modes of communication of individual segments. The conclusion comprises the tests of obtained video records using different vehicles, different natures of driving and the vehicle position at the time of recording.**

***Keywords*—Speed detection; Video processing; Gaussian Mixture Model; DBSCAN clustering; Kalman filter; Optical Flow**

I. Introduction

Thanks to the improvement that is achieved in computer vision and machine learning, we can find application of these methods in many other areas. One of them is traffic monitoring and management system, where the importance is still growing with growing urbanization. This paper aims at speed detection or estimation of vehicles from video stream. Nowadays the most common way to measure speed is by using the radar equipment, therefore it is very important to propose any other concepts like measuring vehicle speed from video stream. Instead of hardware dependency that is problem with radar systems we can use image processing, which is mainly based on software implementation.

There are several papers where many vehicles speed measuring techniques. Most important part of speed measurement system is object detection and tracking are proposed. In [1] and [2], researchers describe system, where frame differencing is used for moving object detection. That method is very efficient in case, where is no movement in the background. More suitable way for object detection is described in [3], [4]. Gaussian mixture model is presented as solution for foreground subtraction with dynamic background. For object tracking optical flow is used that helps to determine not only speed of movement but also direction of the motion. Very interesting project is mentioned in [5], where vehicle speed detection system runs on embedded device Raspberry Pi. Object detection is solved also by Gaussian Mixture model, which proves that this method is suitable also for devices with limited resources. Speed estimation from video frames using corner detection is shown in [6]. For vehicle segmentation also combination of edge detectors, corner detection and morphological operations is used. Another way of using image processing technique to detect speed is shown in [7]. Their goal is to detect speed from single blurred image.

II. Proposed system

As it is mentioned before, we propose system that can deal with speed detection in various conditions (Fig. 1). A real parking place is selected as input, where our goal is to solve influence of weather, illumination and also overlays with other objects as you can see in Fig. 2. Whole system is composed of object detection, object tracking and speed estimation and they are all described in details in next subsections.

A. Input

Input video record was produced by industrial camera Samsung iPOLiSSNP - 5200H with 10fps and 1280x1024

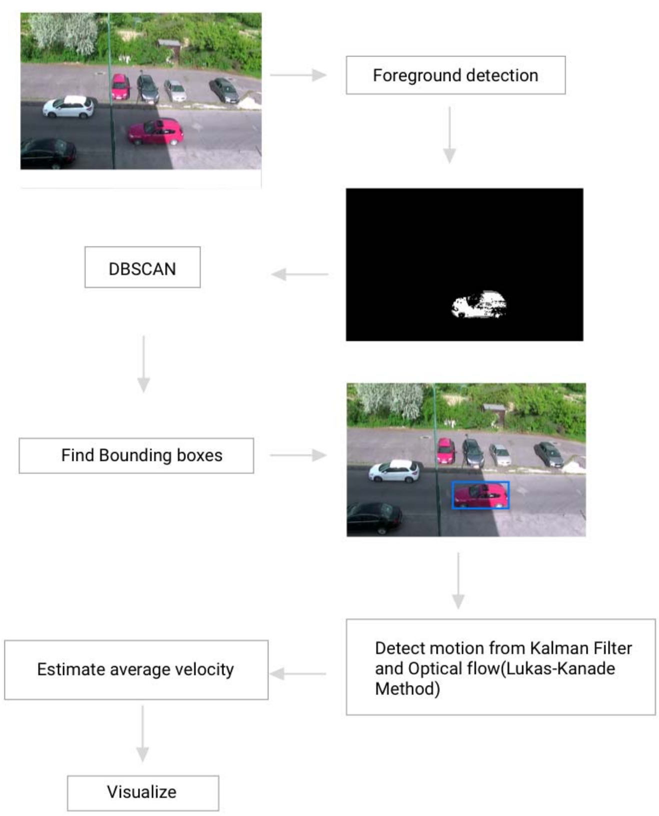

Figure 1. Proposed algorithm.

screen size. Camera is directed upright to the road and beside the road you can also see parking place as is shown in Fig. 2.

B. Object detection

Our approach is based on adaptive background subtraction method called $Gaussian\ mixture\ model$. After each pixel is classified by background subtraction method, segments of foreground points are specified by clusters that are created by $DBSCAN$ ($Density - based\ spatial\ clustering\ of\ applications\ with\ noise$) clustering method and these segments are marked by $Bounding\ boxes$.

1) Gaussian mixture model: (GMM) is a common method for real-time segmentation of moving regions in image sequences. Each pixel is classified as foreground or background based on his representation in the most suitable Gaussian distribution.

a) The method: At any time, t, pixel history (x_0, y_0) is described by equation (1).

$$X_1, ..., X_t = \{V(x_0, y_0, i) : 1 \leq i \leq t\}. \tag{1}$$

The history of each pixel, is modeled by a mixture of K Gaussian distributions(K is based on compute performance, higher values need better hardware).The probability of observing the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta \left(X_t\, \mu_{i,t}, \Sigma_{i,t}\right), \tag{2}$$

where $\omega_{i,t}$, is an estimate of the weight i^{th} Gaussian in the mixture at time t, $\mu_{i,t}$, is mean value, Σ, is the covariance matrix, and where η is a Gaussian probability density function

$$\eta \left(X_t\, \mu_{i,t}, \Sigma_{i,t}\right) = \frac{1}{(2/pi)^{n/2} \Sigma_{i,t}^{0.5}} \exp - \frac{1}{2} (X_t - \mu_{i,t}) \Sigma_{i,t} \left(X_t - \mu_{i,t}\right). \tag{3}$$

$$\Sigma_{k,t} = \sigma_t^2 I. \tag{4}$$

Equation (4), assumes that the red, green, and blue pixel values are independent and have the same variances. Figure 3 shows impact how independent RGB color space is helpful especially with low image quality.

For maximization likelihood of observed data GMM uses $Expectation\ Maximization (EM)$ but it would be costly implement it for every pixel so EM is replaced by $On-line$ $K-means$ approximation.

Pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it [8].

2) DBSCAN Clustering: Method relies on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. Given a set of foreground points from Gaussian mixture model, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).

3) Bounding boxes: For better representation of clusters we mark them by circumscribed rectangles that we also call Bounding boxes.

C. Object tracking

For tracking object we use two diametrically different methods, but at the same time they compete each other and deliver good results. Kalman filter is used to avoid the problems with temporary occlusions and Optical flow provides more accurate speed delivery.

1) Optical Flow: or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene [9], [10]. In our proposed method we use Lukas-Kanade algorithm, which is based on three assumptions:

- brightness constancy(assume that the brightness stay the same from frame to frame),
- temporal persistence or "small movements" (temporal increments are fast enough relative to scale of motion),
- spatial coherence (neighboring points in a scene belong to the same surface and have similar motion).

For two dimensional image, the previous assumptions lead to single equation (5) of two unknowns for any given pixel (y component of velocity v and the x component of velocity u).

$$I_x u + I_y v + I_t = 0. \tag{5}$$

Figure 2. Input image.

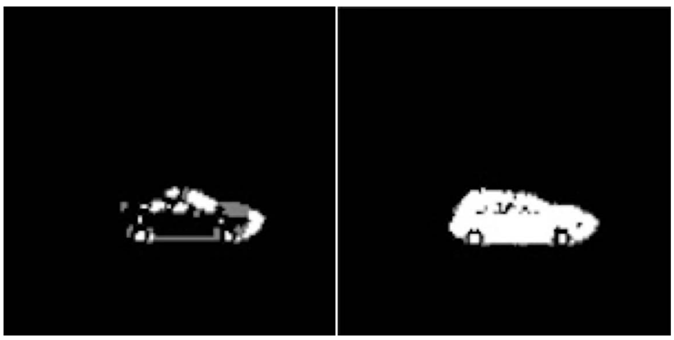

Figure 3. Independent color space difference.

From equation (5) we get

$$\Delta \mathbf{I}^T \mathbf{u} = -I_t, \tag{6}$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}, \tag{7}$$

$$\Delta \mathbf{I} = \begin{bmatrix} I_x \\ I_y \end{bmatrix}. \tag{8}$$

Figure (4) presents mathematical and geometrical details to compute pixel $\mathbf{u}$ [11].

2) Kalman filter: Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by using Bayesian inference and estimating a joint probability distribution over the variables for each timeframe.

The Kalman filter model assumes the true state at time k is evolved from the state at (k - 1) according to

$$\mathbf{x_k} = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{B_k}\mathbf{u}_k + \mathbf{w}_k, \tag{9}$$

where

- $\mathbf{F}_k$ is the state transition model which is applied to the previous state $\mathbf{x}_k$,
- $\mathbf{B}_k$ is the control-input model which is applied to the control vector $\mathbf{u}_{k-1}$,
- $\mathbf{w}_k$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution.

We assume that between the (k - 1) and k timesteps uncontrolled forces cause a constant acceleration, so there is no $\mathbf{Bu}$ [11].

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k. \tag{10}$$

State transition model is fed by $Bounding\ Boxes$ from DBSCAN, so it measure position, width and height of $Bounding\ Boxes$.

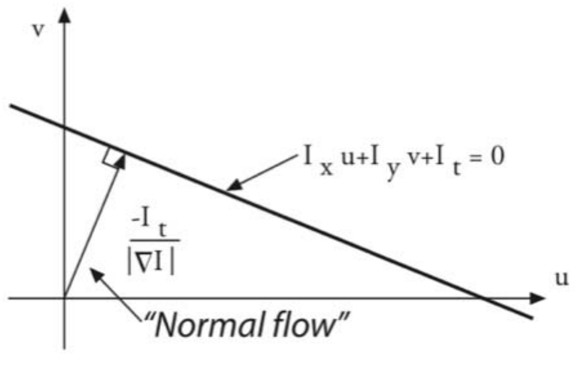

Figure 4. Mathematical and geometric details.

3) Euclidean manager: Euclidean manager is based on communication between $Kalman\ filter$ and $optical\ flow$. Its work is based on three principles:

1) Prediction ($Kalman\ filters$) estimate position based on time difference between frames
2) Merge $Bounding\ Boxes$ with $Kalman\ filters$ by Euclidean distance and update $Kalman\ filters$
3) Delete $Kalman\ filters$ which are out of camera.

Pseudocode for updating $Kalman\ filter$ is as follows:

Require: BouningBox box
  if is first time update **then**
    initialize $Kalman\ filter$
    create $Optical\ flow$ object
  else
    update Kalman filter
    if $Optical\ flow$ object is in $Kalman\ filter$ area **then**
      reinitialize $Optical\ flow$ object
    end if
  end if

D. Speed detection

Kalman filter with Optical flow detects every movement of pixels per second what is also representation of motion of vehicle. For determination of speed, method needs to know weight of the pixel. Paper [4] shows general information about speed determination by dividing real width of double-lane national road with pixel representation.

$$k = \frac{8.5m}{640px}. \tag{11}$$

Then, the speed of the target is computed by the average speed of all points in the target. The speed of the target is given by:

$$m_{ave} = \frac{1}{r}\sum_{n=1}^{w} m_p, \tag{12}$$

where r denotes the number of points in the moving target,m_p denotes the optical flow geometry value of the point of target,m_{ave} denotes the optical flow geometry value of the block. Consequently, the vehicle velocity is given as follows:

$$V = km_{ave}. \tag{13}$$

III. Testing

For testing, we used series of videos with predetermined constant vehicle speed. Test data was measured in different seasons and vehicles.

a) Test car 15 km/h.: Car drove constant speed 15 km/h. Figure 5 visualizes output from video analysis. Near third second Optical flow method losts track pixel flow because of overlay with foreground object. Optical flow was reinitialized by Kalman filter and it continues with speed estimate.

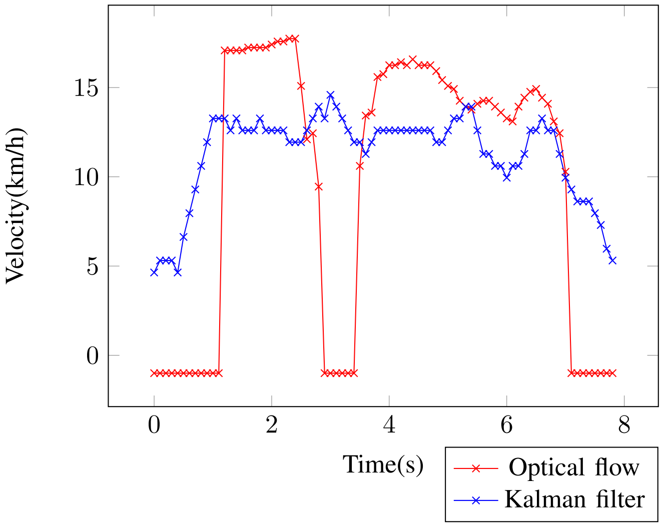

Figure 5. Speed test 15 km/h.

b) Test car 20 km/h.: Car drove constant speed 20 km/h. Figure 6 visualize output from video analysis. Output show various reinitializations of *optical flow* because of low image quality (Fig. 7) and similar overlay with foreground static object.

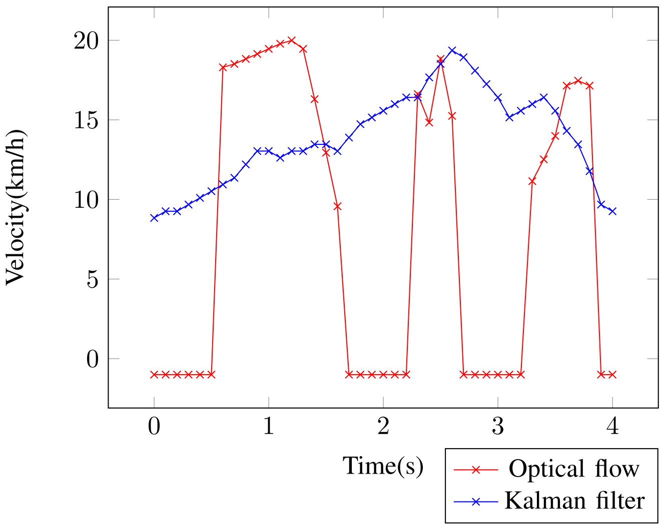

Figure 6. Speed test 20 km/h.

IV. Conclusion

We presented the proposal of vehicle speed measurement system, where we improve optical low method with Kalman filter tracking to solve the problem with overlays with static foreground objects and also improve speed detection. Also foreground detection by Gaussian mixture model was combined with DBSCAN clustering to create more precise object representation. Based on our results we can conclude that combination of optical flow and Kalman filter methods deliver relatively good results even in the case of low image quality produced by industrial camera. Our future research will detect speed from vertically movement by using adaptive weights of pixels and improve DBSCAN segmentation to distinguish every object in cluster of vehicles.

Acknowledgment

The research described in the paper was done within the project No. 1/0867/17 of the Slovak Grant Agency VEGA.

References

[1] C. Pornpanomchai and K. Kongkittisan, "Vehicle speed detection system," in *2009 IEEE International Conference on Signal and Image Processing Applications*, pp. 135–139, Nov 2009.

[2] J. x. Wang, "Research of vehicle speed detection algorithm in video surveillance," in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 349–352, July 2016.

[3] O. Smirg, Z. Smekal, M. K. Dutta, and B. Kakani, "Automatic detection of the direction and speed of moving objects in the video," in *2013 Sixth International Conference on Contemporary Computing (IC3)*, pp. 86–90, Aug 2013.

[4] M. A. Alavianmehr, A. Zahmatkesh, and A. Sodagaran, "A new vehicle detect method based on gaussian mixture model along with estimate moment velocity using optical flow,"

[5] I. Iszaidy, A. Alias, R. Ngadiran, R. B. Ahmad, M. I. Jais, and D. Shuhaizar, "Video size comparison for embedded vehicle speed detection travel time estimation system by using raspberry pi," in *2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, pp. 1–4, Nov 2016.

[6] K. V. K. Kumar, P. Chandrakant, S. Kumar, and K. J. Kushal, "Vehicle speed detection using corner detection," in *Proceedings of the 2014 Fifth International Conference on Signal and Image Processing*, ICSIP '14, (Washington, DC, USA), pp. 253–258, IEEE Computer Society, 2014.

[7] H.-Y. Lin, K.-J. Li, and C.-H. Chang, "Vehicle speed detection from a single motion blurred image," *Image and Vision Computing*, vol. 26, no. 10, pp. 1327 – 1337, 2008.

[8] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, pp. 246–252, IEEE, 1999.

[9] A. Burton and J. Radford, *Thinking in perspective: critical essays in the study of thought processes*. Methuen, 1978.

[10] D. H. Warren and E. R. Strelow, *Electronic Spatial Sensing for the Blind: Contributions from Perception, Rehabilitation, and Computer Vision*, vol. 99. Springer Science & Business Media, 2013.

[11] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

Figure 7. Example of low quality image.