Optical Flow Based Algorithm for Vehicle Counting

Nikola Dojčinović and Jugoslav Joković

Abstract **–General approach for vehicle counting based of videos of traffic flow is presented and discussed. Presented approach provides a framework for different vehicle counting algorithms. Particular implementation of algorithm is provided for conventional traffic monitoring systems. Algorithm reduces number of calculations, neglecting orientation information of moving vehicles, without performing vehicle tracking. Decision criteria is presented for vehicle extraction ensuring reliable vehicle counting in noisy videos.**

Keywords – **Vehicle counting, optical flow, motion detection**

I. Introduction

Traffic congestion is a serious issue confronting many urban centers. This issue is being addressed traditionally by increasing the supply of the roads. Due prohibited costs, such a solution is being abandoned. Instead, contemporary solutions focus on optimizing the throughput of existing roads. Therefore, methods for gathering real-time information about traffic flow are key. First solutions were based on inductive-loop detectors [1] buried underneath roads. Such detectors could only count vehicles travelling over them, without additional information about type and direction. Installation and maintenance of inductive-loop based systems is very expensive and requires traffic stopping.

More recently traffic monitoring systems are based on more promising camera networks. Less disruptive and less costly, camera networks based monitoring systems are very easy to install and to maintain. Key advantage of camera network based over inductive-loop based systems are additional information about traffic flow that camera network system can provide.

One of primary goals of traffic monitoring systems is classification of traffic patterns based on information about traffic flow. Therefore, camera network system must provide enough information about traffic flow. There are two main set of methods for extraction of traffic information from optical information gathered from camera network.

First set of methods directly measures a dynamics aggregated over regions of image in time. Traffic scenes are treated as instances of a dynamic texture [2], i.e. as spatiotemporal image patterns best characterized in terms of the aggregate dynamics of a set of constituent elements, rather than in terms of the individuals.

Another set of methods is based on a combination of segmentation and tracking. Individual objects are being detected and analysed. The general procedure ([3]-[13]) is consisted from the following three steps: (i) motion detection, (ii) tracking of detected vehicles and (iii) combining trajectory information to derive overall description of traffic flow. There are two major problems associated with this approach[14]: (i) segmentation problems due varying environmental conditions (e.g. lightning, strong wind), occlusions and low resolution imagery resulting in small pixel support of vehicle targets and (ii) tracking issues related to correspondence problems and occlusions.

In this paper solution for indirect traffic frequency calculation through temporal vehicle counting is presented. General algorithm for vehicle counting is proposed. Specific contribution of this paper is new method for vehicle counting based on optical flow estimation. New decision criteria for object determination is presented. This method involves no tracking of individual vehicle, and therefore number of calculations is significantly reduced.

II. General Description of Vehicle Counting Algorithm

Algorithms for vehicle counting are oriented toward analysis of individual objects extracted from videos of traffic flow. If whole scene is analysed, without of objects extraction, we could only estimate, but could not get exact number of vehicle on the road. One property of vehicle that uniquely distinguishes vehicles on roadway scenes is their motion. All non-moving objects on the scene can be classified as non-vehicle. In practice, there are non-moving vehicles on the roadway (e.g. stopped or parked vehicle), but they have no significance for traffic frequency estimation because they have no direct influence on traffic congestion.

Motion of the vehicles in the scene is best described with motion vector, because it contains information both of direction and displacement of the vehicles. In literature, there are several methods for motion vector estimation between consecutive frames. Block matching set of methods are based on simple and straightforward, and yet very efficient, algorithm presented in [16]. Fundamental idea of block matching is to find non-overlapped, equally spaced, fixed size, small rectangular blocks of image that match best by some predefined matching criteria. Methods for motion vector estimation from block matching set defer one to another mostly in procedure for searching of best matching blocks. Various new methods are proposed, among them coarse fine three-step search [17], conjugate direction search [18]. Some of them are using multi-resolution block matching [19] or subsampling [20]. Although there are significant limitations of block matching [21], it has been by the most popularly utilized motion estimation technique in video coding and it has been adopted by major video coding standards (ISO MPEG-1 and MPEG-2 [22], ITU H.261, H.263 and H.264 [23]).

Optical flow methods provide relatively more accurate motion vector estimation than block matching. Starting point of optical flow estimation is modelling of typical motion recording system. 2D representation of 3D scene over parametric transformation is used. This model is valid only if difference in depth caused by motion is small relatively to the distance of object from camera objective. Another prerequisite

Authors are with Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: dojca@elfak.rs , jugoslav.jokovic@elfak.ni.ac.rs

for good modelling is constant scale of object in the scene. Change of object dimensions (object re-scaling) is detected as motion, e.g. object dimensions increasing are detected as depth reduction, moving toward camera, and vice versa, object dimension decreasing is detected as moving from the camera. Let I(x, y, t) be intensity level of pixel in frame t, positioned in x-th row and y-th column of image matrix. If displacement induced in pixel (x, y) is presented as pair (p(x, y, t), q(x, y, t)), assumption of grey level constancy between two frames :

$$I(x+p(x,y,t), y+p(x,y,t), t+1) = I(x,y,t) \tag{1}$$

If presumption that movement is small relatively to dimension of the frame is used, it is considered infinitesimal, so its valid expanding of left side of equation (1) to its first order Taylor expansion around (x, y, t) and neglecting all nonlinear terms yields:

$$I(x+p, y+q, t+1) = I(x,y,t) + pI_x + qI_y + I_t \tag{2}$$

where

$$I_x = \frac{dI(x,y,t)}{dx}, \quad I_y = \frac{dI(x,y,t)}{dy}, \quad I_t = \frac{dI(x,y,t)}{dt}, \tag{3}$$
$$p = p(x,y,t), \quad q = q(x,y,t)$$

Equations (1) and (2) yields the well-known Horn-Schunck constant [24]:

$$pI_x + qI_y + I_t = 0. \tag{4}$$

We look for motion (p , q) which minimizes the error function at frame t in the region of analysis R:

$$Err^{(t)}(p,q) = \sum_{(x,y)\in R} (pI_x + qI_y + I_t)^2 \tag{5}$$

Now we perform the error minimization over the parameters of one of the following motion models [25]:

1. **Translation model**: Motion is described with 2 parameters, $p(x, y, t) = a$, $q(x, y, t)=d$, where is assumed that the entire object have a single translation. First order derivate minimization is performed by setting $Err^{(t)}(p, q)$ derivatives with respect to a and d to 0. We get two linear equation with two unknowns, *a* and *d*. In related papers [26] and [27], every small window is assumed to have a single translation.
2. **Affine model:** Motion is described with 6 parameters, $p(x, y, t) = a + bx +cy$, $q(x, y, t)=d + ex + fy$. Using derivation of $Err^{(t)}(p, q)$ with respect to the motion parameters and setting to zero yields six linear equations with six unknowns, *a, b, c, d, e, f* [26][28]
3. **Model of planar surface moving (a pseudo projective transformation):** Motion is described with 8 parameters [26][29], $p(x, y, t) = a + bx +cy + gx^2 +hxy$, $q(x, y, t)=d + ex +fy +gxy + hy^2$. In this case we have eight linear equations with unknowns, *a, b, c, d, e, f, g* and *h*.

Depending of motion model and minimization, optical flow estimation algorithms differ in number of calculations and accuracy of estimation. Therefore, choosing a right motion model is of great significance. If it is possible to predict which motion model will suit particular motion, it is possible to significantly reduce number of calculations, without estimation accuracy loss. Model of planar surface is most general model and will give accurate results for every motion. But, in cases when motion can be described accurate enough with translation model, it is not needed to use more complex ones. It requires much more calculation to solve system of eight linear equations with 8 unknowns (Model of planar surface) than system of two linear equations with two unknowns (Translation model).

If complex motions are analysed, like 3D motions or translation movement captured with moving camera, it is more appropriate to use complex motion models. In literature, there are frameworks for motion vector estimation for all complex model of movement. In [26] is described a hierarchical framework for the computation of motion information. Framework consists from “global model that constrains the overall structure of the motion estimated, a local model that is used in the estimation process, and a coarse-fine refinement strategy”, with application to specific examples.

Number of calculation needed for optical flow estimation directly affects to algorithm temporal properties. For whole algorithm it is essential to count vehicle in real time. This means that optical flow must be estimated in less than time between two frames, so post-processing and analysis can be done.

As intermediate result, matrix of image blocks (pixels) which represent inter-frame calculated movement vectors occurs. Movement vectors are result of object movement in the scene. However, some of movement vector are false positive. Their origin from moving objects which are not vehicles (birds, trees, etc.) and/or noise. To save further computations, false positives are removed with adaptive threshold and filtering.

As early stated, motion vectors are used for extraction of objects from background, and it is a common practice [22][23]. Object is considered as group of neighbourhood image blocks (pixels) with common motion vector property. Depending on particular application, different motion properties are used for object determination. In following section one example of determination property selection will be shown.

Detected objects are than tracked trough the scene and counted. Simple scenes with one roadway do not require tracking of vehicle for counting. In complex scenes, like crossroads, roundabout or roadway intersections, vehicles has to be tracked in order not to be counted more times than one. Tracking algorithms are beyond the scope of this paper.

III. Implementation of Vehicle Counting Algorithm

Regarding to exposure in previous section, general model of system for implementation of vehicle counting algorithm is presented (fig 1). Crocked blocks are optional. After video acquisition, pre-processing is used to optimize video for motion detection (stabilization, adjusting, reconstruction, etc). Role of post-processing block is to filter out small artefacts from false positive motions. Object tracking is, also, optional, depending on previous information about vehicle movement orientation or nature of the scene. Particular implementation is optimized for vehicles counting on straight road. Presumption is that traffic flow is oriented from top of the scene to scene bottom, or vice versa. Cause for this presumption is common positioning of cameras in traffic monitoring systems, above the road (fig 2).

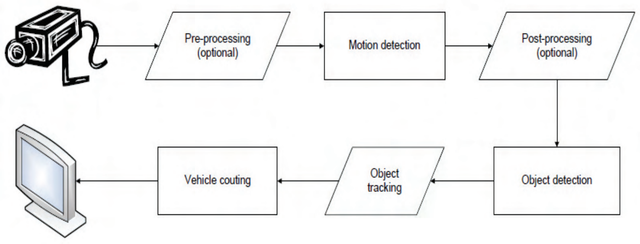


Fig. 1. Scheme of vehicle counting system

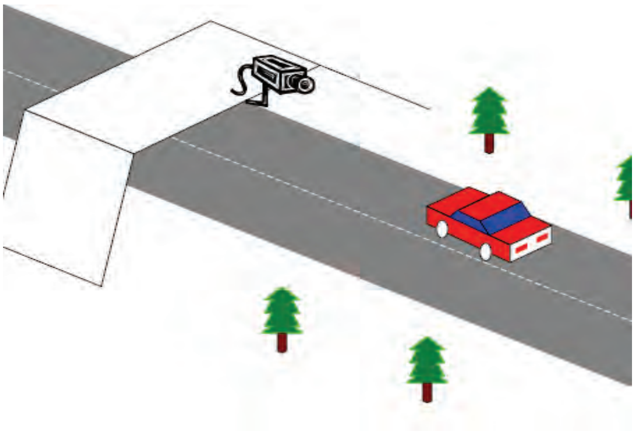


Fig. 2: Common camera positioning

By choosing appropriate motion vector estimation method, curtain facts must be considered. Minimizing number of calculations, motion vector are estimated over smallest possible regions, both in spatial and temporal domain. However, small image blocks carry little motion information, and therefore such motion estimations are inaccurate. Increasing of temporal region to two or more frames improves accuracy of calculated motion vectors. Simple increasing of spatial region can include more than one motion vector in single region. Vehicles have only one motion vector, so we choose spatial region not to contain two vehicles. In other words, region of motion vector estimation must be smaller than smallest vehicle in video.

Fact that vehicle has only one motion vector implies usage of simplest motion model (translation model). In order to reduce number of calculations, Horn&Schunk method is chosen[24].

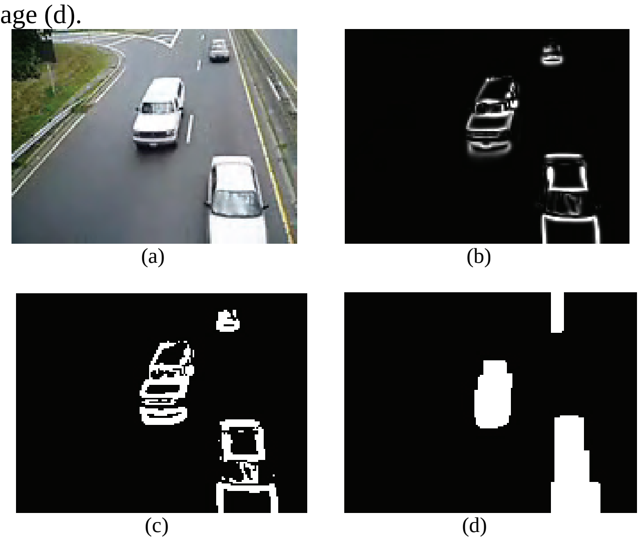
In post-processing, matrix of amplitudes of motion vectors is calculated. Considering presumed orientation of vehicles, only information of motion vector intensity is used for segmentation. Threshold is calculated as spatial mean square value in current frame

$$T_{msv} = k\sqrt{\frac{\sum_{(x,y)\in t} I(x,y)^2}{numel(t)}} \quad (6)$$

in frame t, where *numel(t)* in number of blocks (pixels) in frame t and k scaling parameter. Segmentation over threshold calculated this way is subject to bad segmentation in frames where there are no vehicles in the scene. Absolutely small movements, but relatively greater than threshold, will be false segmented. Therefore, it is needed to insert some inertia in threshold calculation, by applying temporal threshold calculation, over past few frames. Let $T_{msv}(t)$ denotes threshold calculated in frame t, and let n be number of past frames over which calculation is done. Spatiotemporal threshold can be, then, calculated as:

$$T_{temp} = \sqrt{\frac{\sum_{i=0}^{n} T_{msv}(t-i)^2}{n}} \quad (7)$$

In Figure 3 are presented 40th frame on sequence “viptraffic.avi” (a)(standard Matlab sequence), motion vectors image (b), segmented image with spatiotemporal threshold calculated over 3 consecutive frames (c) and post-processed image (d).



(a)

(b)

(c)

(d)

Fig. 3: Intermediate representations of 40th frame of original sequence “viptraffic.avi”(a), motion vector representation (b), segmented image (c) and post-processed image (d).

After segmentation, intermediate results are post-processed in order to remove false positive motions and to fill in space within closed contours (fig 3.c). Amount of information infects accuracy of motion vector estimation. As much information are contained in region, estimation is more accurate. High frequency details, like edges, contain information about contour of the objects, and therefore their motion vectors are estimated more accurate. Motion vectors of uniformly distributed regions are estimated less accurate, so that in segmented image object contours are presented. Performing operations of morphological erosion and closing, small artifact are filtered out and contours are filled in (fig 3.d).

In bimodal image (fig 3.d) objects are detected, as groups of neighboring positively segmented blocks (pixels). Not all detected objects are vehicles. Performing post-processing as described in previous section, small artifacts are filtered out, but procedure will have opposite impacts on large false detected blocks. This blocks are filled in, like vehicles contours (fig 4). What differs them from vehicle objects is their spatial distribution. While vehicle objects are homogeneous, consistent, non-vehicle objects are jagged, spatially displaced from centers (fig 4). Therefore, decision criteria based on object homogeneity is employed for object classification on vehicle objects and non-vehicle objects.

For every object unique property of homogeneity, h_c, is calculated as ratio of object area and area of square around the object (8). If coordinates of up, left corner of the square around the object are (x_1, y_1) and down, right coordinates are (x_2, y_2)

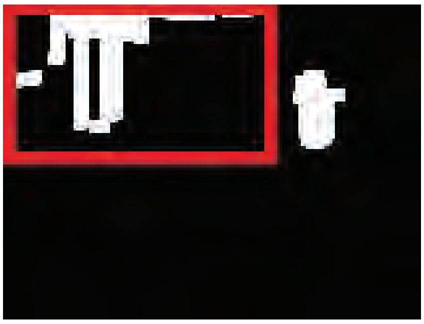
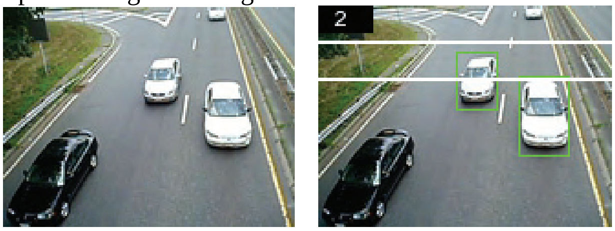


Fig. 4: Red square non-vehicle object and vehicle object right from it

$$h_c = \frac{area(object)}{abs[(x_2-x_1)(y_2-y_1)]} \tag{8}$$

with maximum value 1 (squared objects).

If h_c is greater than ratio threshold, object is considered as vehicle. Vehicles are counted over counting area (fig 5). Area is defined with start and stop line. Orientation is normal to orientation of the traffic flow and width is set to be greater than greatest expected object displacement, so that no vehicle can pass through counting area uncounted.



(a) (b)

Fig. 5: original frame (a), counting area with counter of number of vehicles currently present in it (b)

IV. Conclusion

Vehicle counting based of video information of traffic flow happens to be more accurate, less expensive and easily applicable, than traditional inductive-loop based. Proposed algorithm, provided for commonly used, standard traffic monitoring systems, counts vehicles in real time, neglecting motion orientation, and therefore significantly reduces number of calculations. In case of noisy videos, decision criteria ensures reliable vehicle counting and enables vehicle shape information. Further calibration can enable vehicle velocity estimation. Although it is not a optimal solution, due great cost of computation, this algorithm has provided a solid framework for further development.

References

[1] Traffic Detector Handbook, page 338. Institute Transportation Engineers, 1990.

[2] D. Chetverikov and R. Peteri. A brief survey of dynamic texture description and recognition. In CORES, pages 17–26, 2005.

[3] D. Koller, K. Daniilidis, and H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. IJCV, 10(3):257–281, 1993.

[4] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In ICPR, pages I:126–131, 1994.

[5] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real time computer vision system for measuring traffic parameters. In *CVPR*, pages 495–501, 1997.

[6] A. Lipton, H. Fujiyoshi, and R. Patil. Moving target classification and tracking from real time video. In WACV, pages 8–14, 1998.

[7] T. Tan, G. Sullivan, and K. Baker. Model-based localization and recognition of road vehicles. IJCV, 27(1):5–25, 1998.

[8] R. Cucchiara, M. Piccardi, and P. Mello. Image analysis and rule-based reasoning for a traffic monitoring system. Trans. ITS, 1(2):119–130, 2000.

[9] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. Trans. ITS, 1(2):108–118, 2000

[10] Y. Jung, K. Lee, and Y. Ho. Content-based event retrieval using semantic scene interpretation for automated traffic surveillance. Trans. ITS, 2(3):151–163, 2001

[11] B. Maurin, O. Masoud, and N. Papanikolopoulos. Monitoring crowded traffic scenes. In ICITS, pages 19–24, 2002

[12] D. Magee. Tracking multiple vehicles using foreground, background and motion models. IVC, 22(2):143–155, 2004

[13] A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. CirSysVideo, 15(4):575–584, 2005

[14] K.G. Derpanis and R.P.Wildes. Classification of Traffic Video Based on a Spatiotemporal Orientation Analysis

[15] Yun Q. Shi and Huifang Sun. Image and Video Compression for Multimedia Engineering, 2nd ed, CRC Press, 2008

[16] J.R. Jain and A.K. Jain, Displacement measurement and its application in intraframe image coding, *IEEE Transaction on Communications*, COM-29, 12, 1799-1808, December 1981

[17] T. Koga, K. Linduma, A. Hirano, Y. Ilijima and T. Ishiguro, Motion compensated interframe coding for video conferencing, Proceeding of NTC'81, New Orleans, LA, pp. G5.3.1-G5.3.5, December 1981.

[18] R. Srinivasan and K.R.Rao, Predictive coding based on efficient motion estimation, Proceedings of ICC, pp. 521-526, May 1984

[19] D. Tzovras, M.G. Strintzis and H. Sahinoulou, Evaluation of multiresolution block matching techniques for motion and disparity estimation, Signal Processing: Image Communication, 6, 56-67, 1994

[20] M. Bierling, Displacement estimation by hierarchical block matching , Proceeding of Visual Communication and Image Processing, SPIE 1001, pp. 942-951, 1988.

[21] S. Lin, Y.Q. Shi and Y.Q. Zang, An optical flow based motion compensation algorithm for very low bitrate video coding, Proceeding of 1997 IEEE International Conference on Acoustics, Speech and Signal Processing, Munich, Germany, pp.2869-2872

[22] www.iso.org

[23] www.itu.int

[24] B.K.P. Horn and B.G. Schunck, Determining optical flow, Artificial Intelligence, 17:185-203, 1981

[25] M. Irani, B. Rousso and S. Peleg, Computing Occluding and Transparent Motions, International Journal of Computer Vision, 1994.

[26] J.R. Bergen, P. Anandan, K.J. Hanna and R. Hingorani, Hierarchical model-based motion estimation, In European Conference on Computer Vision, pp. 237-253, Santa Margarita Ligura, May 1992.

[27] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Image Understanding Workshop, pp. 121-130, 1981.

[28] J.R. Bergen, P.J. Burt, R. Hingorani, P. Jeanne and S. Peleg, Dynamic multiple-motion computation. In Y.A. Feldman and A. Bruckstein, editors, Artificial Intelligence and Computer Vision: Proceeding of the Israeli Conference, pp. 147-156, Elsevier, 1991.