Number IV, Volume V, December 2010

RAILWAY CROSSING MONITORING BY CAMERA USING OPTICAL FLOW ESTIMATION

Zdeněk Šilar[1]

Summary: The content of this article is description of the issue of railway crossing monitoring using methods of optical flow estimation. The methods were programmed either directly in Matlab or using Simulink Blocksets.

Key words: Railway crossing monitoring, image processing, optical flow, Matlab, Simulink

1. INTRODUCTION

For monitoring the situation on a railway crossing it is also convenient, besides the current safety installations, to take use of image processing methods [1]. At present namely the two following ways are used for monitoring objects in the defined zone of interest:

- **Objects monitoring using background estimation** [2]

  The method is based on subtraction of a background model from the real image. There are two principal disadvantages of this method:

  a) the need of the image without foreground and b) a high sensitivity to lightening variations. Moreover, in the case of dynamic sequences the background changes during the motion and during the calculation as well (it does not hold for a static camera).
- **Objects monitoring using optical flow**

  For the optical flow computation the image without foreground is not necessary. Moreover we obtain more information (the motion direction and velocity) useful in further phases of the image processing. The main disadvantage of these methods is a higher computation complexity at the expense of the accuracy.

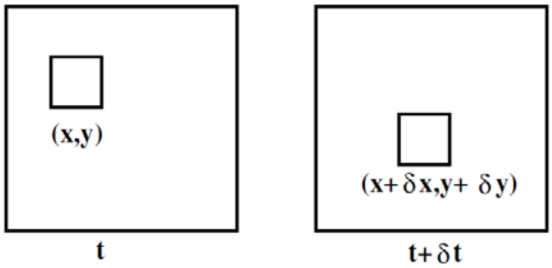
In the following chapters the concept of the optical flow will be explained and the main methods for the optical flow computation from acquired data (static as well as dynamic) will be described. Afterwards the methods will be implemented in Matlab and Simulink environments for image of railway crossing area processing.

2. OPTICAL FLOW

The optical flow describes the direction and time rate of pixels in a time sequence of two consequent images. A two dimensional velocity vector, carrying information on the direction and the velocity of motion is assigned to each pixel in a given place of the picture. The optical flow could be used in situations when an observer (camera) is static and objects in the picture are moving. This is the case of railway crossing monitoring.

[1] Ing. Zdeněk Šilar, University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Information Technology, Studentská 95, 53210 Pardubice, Phone: +420 466036761,
E-mail: zdenek.silar@upce.cz

For making computation simpler and quicker we can transfer the real world [*3D+t*] objects to the [*2D+t*] case (figure 1).



Source: [3]

Fig. 1 – Displacement of 2-D object

Then we can describe the image by means of the 2-D dynamic brightness function of location and time *I(x, y, t)*. Provided that the change of brightness intensity does not happen, in the neighbourhood of the displaced pixel, we can use the following expressions [3]

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

Using Taylor series for the right hand part of the equation (1) we obtain

$$I(x+\delta x,y+\delta y,t+\delta t)=I(x,y,t)+\frac{\partial I}{\partial x}\delta x+\frac{\partial I}{\partial y}\delta y+\frac{\partial I}{\partial t}\delta t+H.O.T \tag{2}$$

From expressions (1) and (2) and with neglecting higher order terms (H.O.T.) we get

$$\frac{\partial I}{\partial x}\delta x+\frac{\partial I}{\partial y}\delta y+\frac{\partial I}{\partial t}\delta t=0 \tag{3}$$

after modification
$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t}+\frac{\partial I}{\partial y}\frac{\delta y}{\delta t}+\frac{\partial I}{\partial t}\frac{\delta t}{\delta t}=0 \tag{4}$$

or
$$\frac{\partial I}{\partial x}v_x+\frac{\partial I}{\partial y}v_y+\frac{\partial I}{\partial t}1=0 \tag{5}$$

where v_x and v_y are components of pixel rate (optical flow) and $\frac{\partial I}{\partial}$ are derivations of the pixel brightness intensity in (x,y,t).

Using common abbreviation of partial derivations the equation (5) after a slight modification holds

$$I_x\cdot v_x+I_y\cdot v_y=-I_t \tag{6}$$

or in a vector representation

$$(I_x,I_y)\cdot(v_x,v_y)=-I_t \tag{7}$$

or formally
$$\nabla I\cdot\vec{v}=-I_t \tag{8}$$

where ∇I is so-called the spatial gradient of brightness intensity and $\vec{v}$ is the optical flow (velocity vector) of image pixel and I_t is the time derivation of the brightness intensity.

The formula (8) is keynote for optical flow calculation and is called 2-D Motion Constraint Equation or Gradient Constraint [3]. It represents one equation with two unknown quantities (the aperture problem).

The formula (8) expresses only the component of optical flow in the gradient direction, marked as a normal component $\vec{v}_n$, which is determined by expression

$$\vec{v}_n = v_n \cdot k \tag{9}$$

where v_n is a scalar normal-line velocity, k is a normal-line direction and both components are determined on the basis of relations (7) and (8)

$$v_n = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}} = \frac{-I_t}{\|\nabla I\|_2} \tag{10}$$

and

$$k = \frac{\left(I_x, I_y\right)}{\|\nabla I\|_2} \tag{11}$$

Methods for optical flow estimation

Optical flow estimation is computationally demanding. At present there are several groups of methods for its calculation [4]. All the methods come from the expression (1), consequently from the presumption of conservation of brightness intensity. In this article our interest is concentrated to the following groups and methods:

- **Block-based methods** – are based on minimizing Sum of Absolute Differences (***SAD***) or maximizing Normalized Cross-Correlation (***NCC***).
- **Differential methods** – they issue directly from the formula (1). The result is a set of differential equations for image brightness intensity. The most frequently used methods pre-prorammed in Matlab and Simulink are the **Horn-Schunck** and **Lucas-Kanade** ones.

3. BLOCK-BASED METHODS

Regions searching (also Region-based Matching) uses correlation functions that can determine the degree of similarity of two different images. Thanks to the continuity of correlation functions it is possible to use techniques for searching their global minimum (***SAD***) or maximum (***NCC***).

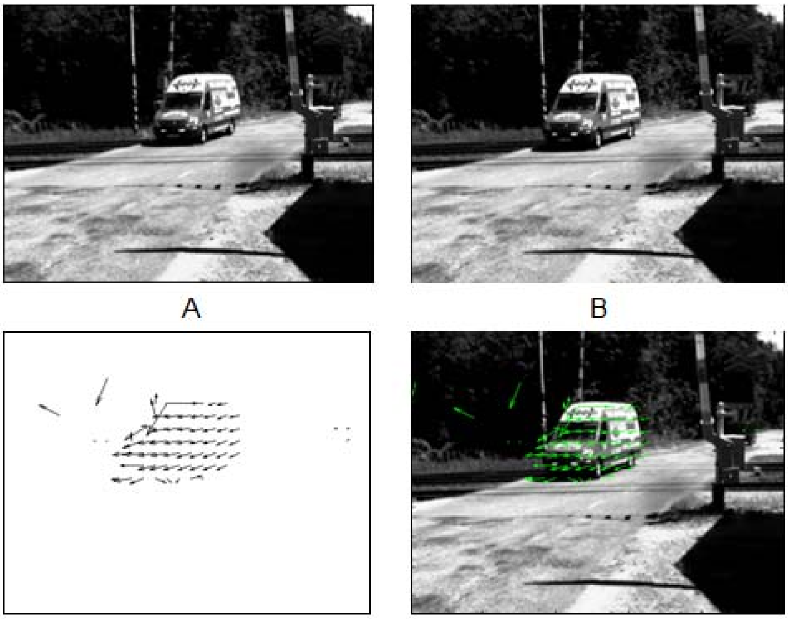
For the optical flow calculation by means of the *Matlab* the correlation method ***SAD*** was used in the form

$$SAD(x_A, y_A, x_B, y_B) = \sum_{i=0}^{n}\sum_{j=0}^{m}\left|I_A(x_A+i, y_A+j) - I_B(x_B+i, y_B+j)\right| \tag{12}$$

where I_A a I_B are the brightness intensities of images *A, B; n* and *m* are macroblock dimensions; *x* and *y* are coordinates of positions of the both images.

The image is practically divided into regions in which the optical flow is determined – searched through by means of the defined macroblocks. The method SAD works so that it calculates the sum of absolute differences of pixels brightness between the compared blocks of the both shifted images. The result then represents the similarity between blocks. The smaller this value is the more similar the blocks are. Everything is performed over the whole searched region (selective window) and so-called the global minimum is determined.

Figure 2 shows the optical flow vectors calculated by comparing two images using the SAD algorithm.



Source: Author

Fig. 2 – The optical flow calculated from the sequence of the two static images (A, B) by means of algorithm *SAD* (9) programmed in Matlab environment

Another option for the optical flow estimation is determination of global maximum by the method of normalized cross correlation (*NCC*). The algorithm is implemented in Matlab using *xcorr2* function. This is a 2-D cross correlation which effectively works directly with image matrixes (without the necessity of programming sums by means of cycles).

The described methods solve the optical flow calculation between the two static images by so-called ”rough force”. Algorithms are accurate and it is possible to set more parameters (macroblock size, searched region etc.). Their main disadvantage is a great calculation and time demand, which could be possibly partially reduced by optimizing steps or by using built-in functions (*xcorr2*) in Matlab environment.

4. DIFFERENTIAL METHODS

The differential methods remove disadvantages of above mentioned correlation methods. The optical flow determination is solved by the calculation of partial derivations of the image signal. The methods again issue from the presumption of gradient constraint (8).

There are the two most used methods[2] implemented in Simulink (extension of Matlab) environment:

[2] Both the methods are already programmed by third-parties, even for Matlab.

- Lucas-Kanade
- Horn-Schunck

4.1 Lucas-Kanade method

Lucas and Kanade [5] introduce the error term ρ_{LK} for each pixel. This one, according to the following relation, is calculated as the sum of the weighted smallest squares of gradient constraint (8) in close ambient of the pixel.

$$\rho_{LK} = \sum_{x,y\in\Omega} W^2(x,y)\left[\nabla I(x,y,t)\cdot\vec{v} + I_t(x,y,t)\right]^2 \tag{13}$$

where Ω is neighbourhood of the pixel; *W(x,y)* are weights allocated to individual pixels in Ω (typically 2-D Gaussian coefficients).

To find a minimal error it is necessary to compute derivation of the error term ρ_{LK} by individual components of velocity and put the result equal to zero.

$$\frac{\partial\rho_{LK}}{\partial v_x} = 2\sum_{(x,y)\in\Omega} I_x(x,y)W^2(x,y)\left[I_x(x,y)v_x + I_y(x,y)v_y + I_t(x,y)\right] = 0 \tag{14}$$

$$\frac{\partial\rho_{LK}}{\partial v_y} = 2\sum_{(x,y)\in\Omega} I_y(x,y)W^2(x,y)\left[I_x(x,y)v_x + I_y(x,y)v_y + I_t(x,y)\right] = 0 \tag{15}$$

After adjustments and transfer to a matrix form the expression for the optical flow calculation is as follows

$$\vec{v} = \left[A^T W^2 A\right]^{-1} A^T W^2 \vec{b} \tag{16}$$

where for *N* pixels ($N = n^2$, for *n* x *n* of Ω neighbourhood) and $(x_i, y_i)\in\Omega$ in time *t* holds

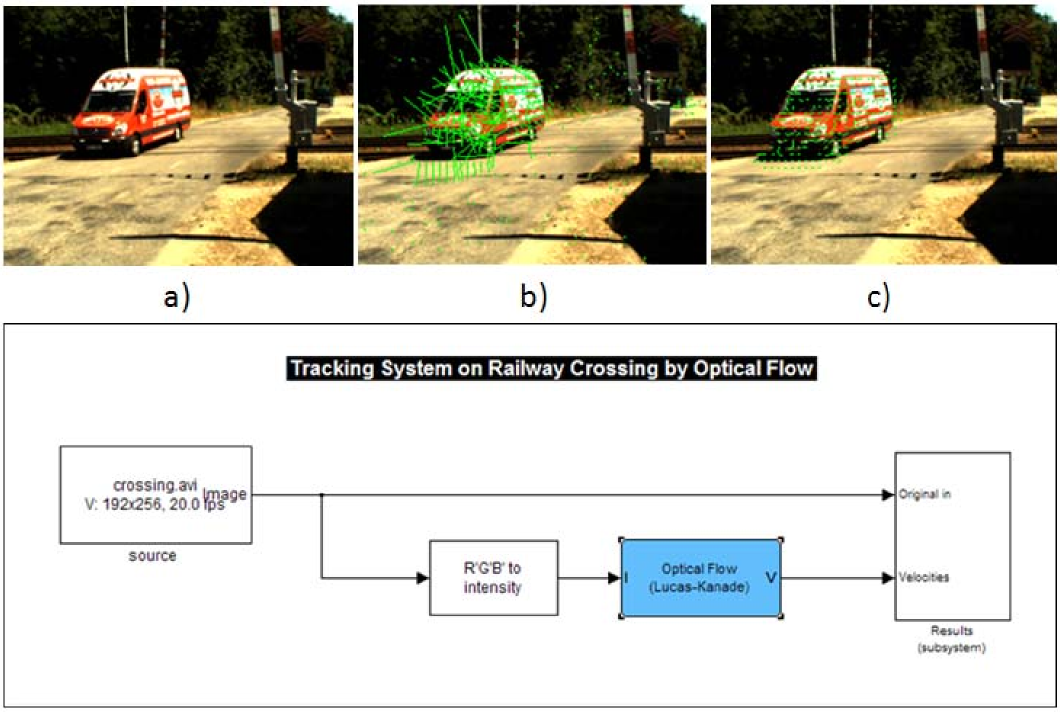
$$A = \left[\nabla I(x_1,y_1),\ldots,\nabla I(x_N,y_N)\right]$$

$$W = diag\left[W(x_1,y_1),\ldots,W(x_N,y_N)\right]$$

$$\vec{b} = -(I_t(x_1,y_1),\ldots,I_t(x_N,y_N)) \tag{17}$$

So we will obtain the resultant velocity for one pixel by solution of the system (16). Instead of the calculation of the sums, the convolution by means of Gaussian or other separable filter is used to reduce the algorithm complicacy.

Simulink environment enable to process the whole dynamic sequences with the help of blocks from the library *Video and Image Processing Blockset.* In the figure 3 the module for calculation and optical flow display (from acquired AVI file) by Lucas-Kanade method implemented in Simulink by the *Optical Flow Block* is applied.



Source: Author

Fig. 3 – *Optical Flow Block* with the chosen Lucas-Kanade method; a) Original video, b) Result by Difference filter, c) Result by Derivative of Gaussian

It is evident from Fig. 3c) that the variant Lucas-Kanade with Gaussian separable filter is more suitable.

4.2 Horn-Schunck method

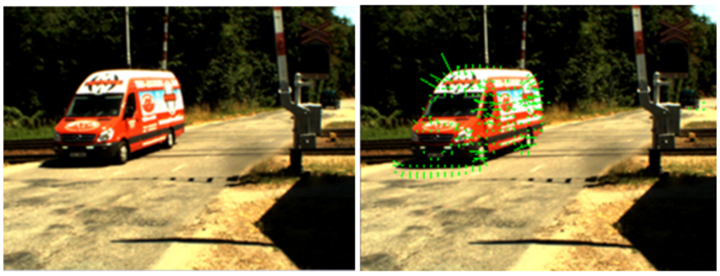
Horn and Schunck [6] issue from Lucas-Kanade method (expression 13). In addition to the gradient constraint they add another error term (called the global term of smoothing) for the limitation of too great changes of optical flow components (v_x,v_y) in the Ω. The minimization of the total error ρ_{HS} is then given by the relation

$$\rho_{HS} = \int_D (\nabla I \cdot \vec{v} + I_t)^2 + \lambda^2 \left[\left(\frac{\partial v_x}{\partial x} \right)^2 + \left(\frac{\partial v_x}{\partial y} \right)^2 + \left(\frac{\partial v_y}{\partial x} \right)^2 + \left(\frac{\partial v_y}{\partial y} \right)^2 \right] dxdy \qquad (18)$$

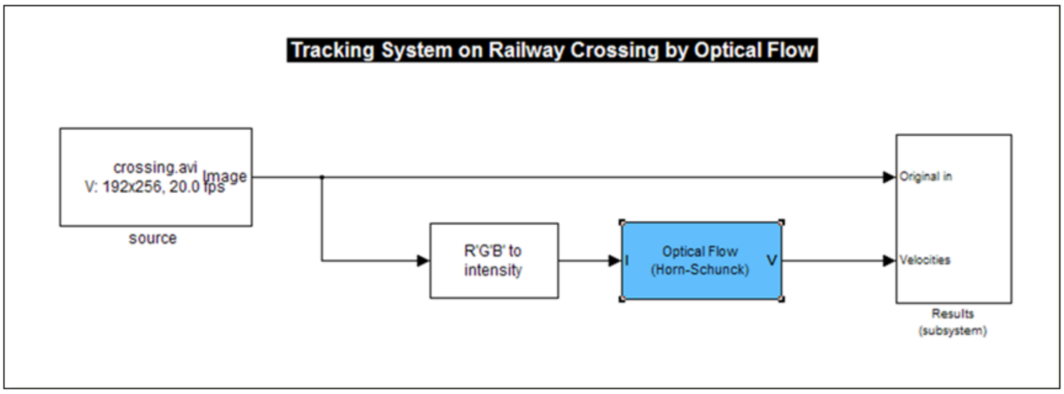
where D (domain) is the region of the whole image, λ expresses relative effect of the second added error term (typically λ=1.0).

The relation (18) leads to the system of equations for whose solution it is convenient to use Jacobi or Gauss-Seidel iterative methods [7].

Horn-Schunck method is more accurate but with regard to relatively large number of iterations (in practice there are 10 to 100 steps) it is slower. The results of optical flow calculation by means of *Optical Flow Block* with Horn-Schunck method is shown in the figure 4.



a) b)



Source: Author

Fig. 4 – *Optical Flow Block* with chosen Horn-Schunck method; a) Original video, b) Result; λ =1.0, iterations = 10

5. CONCLUSION

It is possible to detect effectively moving objects in the region of railway crossing by means of the optical flow. It can be performed from the sequence of two static images or over the individual snaps of the whole video stream. The content of this article is an explanation and definition of the concept of optical flow and description of chosen methods for its estimation. The closing part is devoted to implementing and using methods in Matlab and Simulink environments.

Comparing the results the most suitable methods for optical flow estimation appears to be the Lucas-Kanade method using Gaussian separable filter. The results of optical flow estimation for objects detection in the region of railway crossing will be further processed by other progressive methods for the recognition of images like thresholding, mathematical morphology, clustering, techniques for region filtering, SVD decomposition (Singular Value Decomposition) or other technologies of computer vision.

The research of the railway crossing monitoring is a part of the project No. FR - T11/084, supported by the Czech Ministry of Industry and Business.

REFERENCES

[1] Šilar Z. *Moderní obrazové metody pro sledování a kontrolu železničního přejezdu.* Perner´s Contacts, ISSN 1801-674X, Vol. 4, No. 3, pp 226-240, Pardubice, 2009.

[2] Sankari M., Meena C. *Adaptive Background Estimation and object detection applying in Automated visual surveillance.* (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 4, July 2010.

[3] Barron J. L., Thacker N. A. *Tutorial: Computing 2D and 3D Optical Flow.* Last updated 2005-1-20 [cit. 2010-10-19]. Available www: <http://www.tina-vision.net/docs/memos /2004-012.pdf>.

[4] Barron J. L., Fleet D. J. *Performance of Optical Flow Techniques.* International Journal of Computer Vision , Vol. 12, No. 1, pp 43-77, 1994.

[5] Lucas B. D., Kanade T. *An Iterative Image Registration Technique with an Application to Stereo Vision*. DARPA Image Understanding Workshop, pp 121–130, 1981.

[6] Horn B. K. P., Schunck B. G. *Determining Optical Flow*. Artificial Intelligence, No. 17, pp 185–204, 1981.

[7] Hageman, L., Young, D. *Applied Iterative Methods*. Academic Press, New York, 1981.