Sensors & Transducers, Vol. 21, Special Issue, May 2013, pp. 205-210

Sensors & Transducers

http://www.sensorsportal.com

Improved Optical Flow Algorithm for an Intelligent Traffic Tracking System

[1] Xia Yupeng, [2] Hu Feng
[1] School of Mechanical and Electrical Engineering, Shenzhen Polytechnic,
Xili Lake, Shenzhen, Guangdong, 518055, China
[2] Shenzhen Graduate School, Harbin Institution and Technology,
Xili Lake, Shenzhen, Guangdong 518055, China
[1] E-mail: xiayupeng2013@163.com

Received: 9 April 2013 /Accepted: 14 May 2013 /Published: 30 May 2013

Abstract: It is known that to get the contours and segmentations of moving cars is the essential step of image processing in intelligent traffic tracking systems. As an effective way, the optical flow algorithm is widely used for this kind of applications. But in traditional gradient-based approaches, in order to make the data responding to the edges of moving objects expand to the area, which gray level is flat, it needs to keep the iteration times large enough. It takes a large amount of calculation time, and the accuracy of the result is not as good as expected. In order to improve the numerical reliability of image gradient data, Hessian matrix distinguishing, Gaussian filtering standard deviation amending, mean model amending and multi-image comparing, the four algorithms were investigated by applying them to track moving objects. From the experimental results, it is shown that both the calculation convergence speed and accuracy of our methods have greatly improved comparing with traditional algorithms, the feasibility and validity of those methods were confirmed.

Keywords: Computer vision, Image processing, Optical flow, Image grads, Hessian matrix, Gaussian filtering, Mean model.

1. Introduction

Since the conception, optical flow, was first raised, there are a lot of successful applications of it. We call that apparent motion caused by image brightness patterns, Optical Flow Field, who reflects velocity vectors distribution, which three-dimensional objects project to two-dimensional image plane. Then, optical flow field provides us with a lot of very useful information about velocity and three-dimensional motion structure of moving objects and spatial arrangements of the environment. Up to now, optical flow calculation methods are gradient method, regional method, energy method and phase method. In 1981, Horn and his partners [1] brought forward optical flow constraint equation based on gradient. Since then, various advanced ideas and improved versions have emerged. Nagel [2] used weighted matrix to control gradient data for different smooth handling result, and it improved Horn's global smoothness approach. Lucas and Kanade [3] proposed that optical flow is local smoothness. Their method provides an accurate solution for motion estimation, but it requires large searching window to handle large motions. Juan L et al. [4] take 3D Sobel operator instead of normal Sobel operator, which is in common use for computing image gradient. This method improved result accuracy, but increased large calculation. Thus, maybe it is useful in static scene. Shi Rong [5] proposed a method to eliminate errors caused by high-speed motion analysis. Clearly, it is necessary to repair results, when moving object's velocity is too high.

In this paper, the research on the accuracy and reliability of the gradient data during calculation of the optical flow was preformed, based on Horn's global smoothness method primarily. First, the principle of optical flow and its iterative formula were investigated. Then, the reasons of low accuracy and large calculation burden for the calculation results due to the optical flow algorithm were analyzed. Aiming at current existing shortcomings and problems about the gradient method, some approaches were performed to optimize the image gradient data, when calculating optical flow field. Finally, applying these methods to moving vehicles automatically tracking system, the validity of those methods was confirmed.

2. Horn-Schunck Constraints

Suppose that intensity of a point $p(x, y)$ on an image is $f(x, y, t)$ at time t, and when it arrives at a new position, its intensity turns to $f(x+\Delta x, y+\Delta y, t+\Delta t)$ at time $t+\Delta t$. Without regards to light changing, if $\Delta t \to 0$, we get

$$f(x, y, t) = f(x+\Delta x, y+\Delta y, t+\Delta t) \tag{1}$$

Easily, a dynamic image can be represented as a function of position and time permits it to be expressed as a Taylor series. Ignoring high-order terms, it is easily to get

$$\nabla f \bullet (u, v, 1) = 0 \tag{2}$$

where $u=dx/dt$, $v=dy/dt$. The u and v are horizontal and vertical optical flow vectors, and$\nabla f=(\partial f/\partial x, \partial f/\partial y, \partial f/\partial t)$ is gradient vector along x, y and t. Obviously, it is impossible to determinate optical flow field depending on Eq. (2) only, due to the lack of another constraint. Horn and others' optical flow constraint is based on hypothesis, which is intensity unchangeable and global smoothness. Thus, optical flow must meet the following requirements

$$\min\{E = \iint [u_x^2 + u_y^2 + v_x^2 + v_y^2 + \lambda(f_x u + f_y v + f_t)^2] dxdy\} \tag{3}$$

where λ is weighted coefficient, which reflects that how calculation depend on the quality of image data, and can be decided by image noise. Associating Eq. (2) with (3), analyzing Euler equation and dispersing them, we can get

$$u^{n+1} = \overline{u}^n - \frac{f_x \overline{u}^n + f_y \overline{v}^n + f_t}{\lambda + f_x^2 + f_y^2} f_x , \tag{4}$$

where $\overline{u}$, $\overline{v}$ denote average value of neighborhood, which is decided by the case of application. n denotes iteration times, which is decided by required accuracy. When circulation cannot terminate, it must be set a maximal iterative time.

Usually, the flow chart of optical flow calculation is shown by Fig. 1. There are mostly 3 blocks in the chart, which is image preprocessing, parameters initialization and iterative calculation. In this paper, we amend image gradient data before and during iterative calculation, for the sake of computing optical flow as accurate as possible. For requirement of image segmentation, the iterative results were described by a 256 gray-level bitmap.

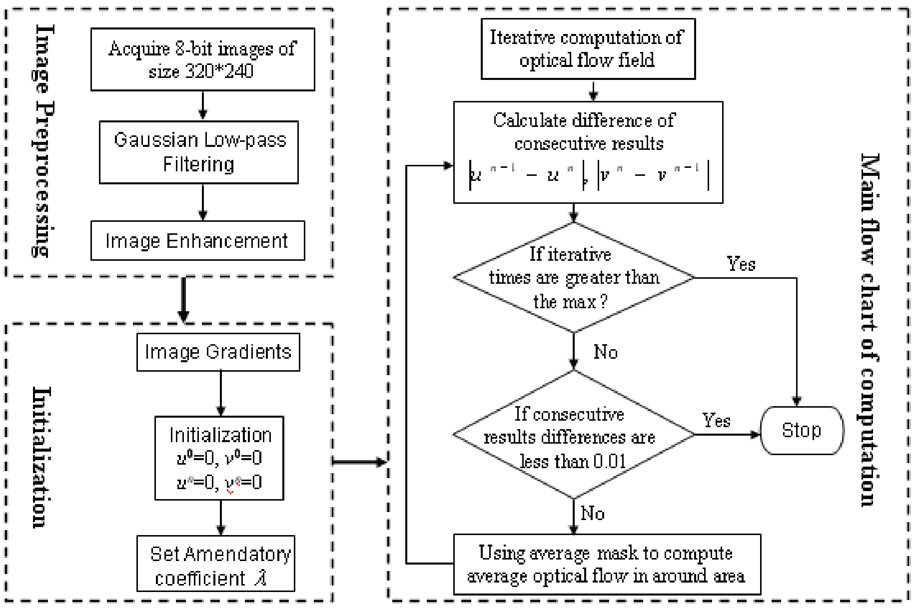


Fig. 1. The flow chart of calculation of optical flow.

The gradient is almost few in the flat area of moving object's gray on image plane. Those can be estimated by computing average value of its bonder data. But this needs a large number of iterative times to keep bonder motion data penetrate to a whole object area. From Eq. (4), we can see that image gradient accuracy and reliability will affect the final accuracy and convergence speed during progress of iterative computation. Hence, we supervise and repair image gradient matrix before and during iterative process, for the sake of getting more accurate optical flow field. In order to illuminate importance of image gradient data, we also compute non-enhancement image, including filtering and sharpening.

2. Methods of Improving Image Gradient Numerical Reliability

3.1. Hessian Matrix

In Yang Chunle's [6] method, they used Hessian matrix to filter image gradient data before calculating optical flow field by using weighted windows local smooth method. Then, they applied Least-squares method for computing optical flow. In this paper, we attempt to use this method in our system for optimizing image gradient data. According to smoothness hypothesis, we can easily obtain the following equation from Eq. (2).

$$\begin{aligned} f_{xx}u + f_{yx}v + f_{tx} = 0\,, \\ f_{xy}u + f_{yy}v + f_{ty} = 0 \end{aligned} \tag{5}$$

And it also can be shown as following

$$\begin{Bmatrix} f_{xx} & f_{yx} \\ f_{xy} & f_{yy} \end{Bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} f_{tx} \\ f_{ty} \end{pmatrix}, \tag{6}$$

Denote $H = \begin{Bmatrix} f_{xx} & f_{yx} \\ f_{xy} & f_{yy} \end{Bmatrix}$, where $\boldsymbol{H}$ is a Hessian matrix of function $f(x, y, t)$. Then, we can eliminate unreliable gradient constraint before optical flow iterative calculation in flow chart 1. The method mainly solves the problem lies in conventional optical flow methods where inaccurate optical flow estimation would always be resulted when unreliable or unreliable gradient constraints exist. From the experimental result, we can see that iterative time is almost cut off 50 %.

3.2. Standard Deviation of Gaussian Curve

There are a lot of noises in images, which is acquired from digital camera by image acquisition. Gaussian filter operator is an effective means to wipe off noises when image processes. Images data are more numerical reliability by filtering. The form of these filters in two dimensions is given by

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\{-\frac{1}{2}(\frac{x^2+y^2}{\sigma^2})\} \tag{7}$$

where σ is the standard deviation of Gaussian curve. When σ approaches infinity, G tends towards a constant function. Usually, σ=1. Bartolini and Pica [7] decided it according to the calculation results of optical flow field as

$$\sigma^2 = \sum_{i=1}^{n} f_i^2 / n - \left(\sum_{i=1}^{n} f_i\right)^2 / n^2\,, \tag{8}$$

In our methods, we calculate optical flow by comparing frame k with frame k-1 and k-2. Because σ is larger, and motion data of moving object border diffuse more enough in images, we hold optical flow results which σ are larger. Flag the method algorithm ***B***.

3.3. Optical Flow Constraint and Smoothness Constraint

It requires optical flow itself as smooth as possible in Horn's smoothness constraint. In Eq. (2), we have ignored high-order Infinitesimal of Taylor expansion. Then, we can define a judgment rule for reliability as following

$$r = \lambda\left|f_x u + f_y v + f_t\right| + u_x^2 + u_y^2 + v_x^2 + v_y^2 \tag{9}$$

where λ is the same as Eq. (3), and u_x, u_y, v_x, v_y are partial differential along x and y of the last time optical flow. As regards this method, in order to obtain more accurate optical flow field, it is possible to validate contentment tolerance of optical flow over again. Acquire 3 consecutive frame images k-2, k-1, k, and separately associate frame k with frame k-1, and k-2 to calculate optical flow. For each pixel of frame k, we calculate optical flow two times on every iterative processing. If ($r_1 \le r_2$), we set $(u, v)=(u_1, v_1)$, or else $(u, v)=(u_2, v_2)$. This approach is similar to algorithm ***B***. Differently, This approach selects and set the best result into formula during every iterative process. For low-speed moving object and ambiguous border image, relative displace position of moving object is less on image plane. It is possible to get good result by using it. Flag it algorithm ***C***.

3.4. Weighted Coefficient

It is obvious that optical flow on edges cannot be decided when objects move along edges, and because gradient changes faster on image corner, there are large errors [8]. When calculating average optical flow in neighborhood area for each pixel, the following usual mask is used.

$$\begin{aligned} \bar{u}^n &= (u_{i,j-1}^{n-1} + u_{i,j+1}^{n-1} + u_{i-1,j}^{n-1} + u_{i+1,j}^{n-1}) / 4, \\ \bar{v}^n &= (v_{i,j-1}^{n-1} + v_{i,j+1}^{n-1} + v_{i-1,j}^{n-1} + v_{i+1,j}^{n-1}) / 4 \end{aligned} \tag{10}$$

Besides, Horn mask is another one in common use. In fact, Image gradient data should be paid more attention to, when calculating corner and edge of images.

Here, we use an advance approach to acquire average value of neighborhood for each pixel, which is focused on edge feature when calculating. On the edge of moving object, we limit that image gradient data should not penetrate from moving object area to background. Thus, we set exponential function e^x as weighted coefficient for each pixel. Of course, in accordance with the actual speed of moving objects and images gradient, we can also choose other functions based of other numbers.

$$\begin{aligned} \bar{u}^n &= \frac{(\mathrm{e}^{-d_{i,j-1}} u_{i,j-1}^{n-1} + \mathrm{e}^{-d_{i,j+1}} u_{i,j+1}^{n-1} + \mathrm{e}^{-d_{i-1,j}} u_{i-1,j}^{n-1} + \mathrm{e}^{-d_{i+1,j}} u_{i+1,j}^{n-1})}{4(\mathrm{e}^{-d_{i,j-1}} + \mathrm{e}^{-d_{i,j+1}} + \mathrm{e}^{-d_{i-1,j}} + \mathrm{e}^{-d_{i+1,j}})}, \\ \bar{v}^n &= \frac{(\mathrm{e}^{-d_{i,j-1}} v_{i,j-1}^{n-1} + \mathrm{e}^{-d_{i,j+1}} v_{i,j+1}^{n-1} + \mathrm{e}^{-d_{i-1,j}} v_{i-1,j}^{n-1} + \mathrm{e}^{-d_{i+1,j}} v_{i+1,j}^{n-1})}{4(\mathrm{e}^{-d_{i,j-1}} + \mathrm{e}^{-d_{i,j+1}} + \mathrm{e}^{-d_{i-1,j}} + \mathrm{e}^{-d_{i+1,j}})} \end{aligned} \tag{11}$$

where $d_{(\cdot,\cdot)}$ is gray-level difference absolute value along x and y of each pixel. Table 1 shows the typical average value template which is obtained by Eq. (11). We take the one above instead of average mask for optical flow of each pixel, to effectively prevent the indiscriminate spread of gradient data. Flag it algorithm *D*.

Table 1. Typical average value template.

0	1/4	0
1/4	1	1/4
0	1/4	0

4. Experimental Result and Discussion

In order to make sure that above approaches are effective, we use several series images of cars about traffic scene to estimate motion parameter and to test system's performance. On our system, we acquire series motion images from a camera, and use above approaches to compute optical flow field for distilling moving objects' contour, and, then, control step-motors to track moving objects on real-time. The image sizes are all 320×240. Images must be preprocessed, such as Gaussian low-pass filtering and Laplace sharpening, before being calculated.

If the calculation error is two times continuous less than 0.01 during iteration, then the results are judged as converge, and calculation would be stopped. The final result is shown with a 256 bitmap. Usually, there are two criterions for estimating computation error of optical flow, which is angle error average value and magnitude error average value, whose definition is as following

$$\begin{aligned} f &= \left\| \vec{u} - \vec{\tilde{u}} \right\| \\ \theta &= \arccos\left(\frac{<\vec{u}, \vec{\tilde{u}}>}{\|\vec{u}\| \bullet \|\vec{\tilde{u}}\|}\right) \end{aligned}, \tag{12}$$

where $\vec{u} = (u, v)$ is real velocity of moving object in image, and $\vec{\tilde{u}} = (\tilde{u}, \tilde{v})$ is optical flow velocity vector. But, in fact, because it is difficult to count the real velocity of vehicles in traffic scene, we used PNSR and iterative times and converge time to evaluate above algorithm. The PSNR is given by

$$\sum_{x=0,y=0}^{w,h} (f_1(x,y) - f_2(x+u(x,y), y+v(x,y)))^2 /(wh))], \tag{13}$$

where *w*, *h* are image's width and height.

In Table 2, the data of the column "non-preprocess image" are results calculated according to Horn method from source images without enhancing.

Table 2. Calculation results.

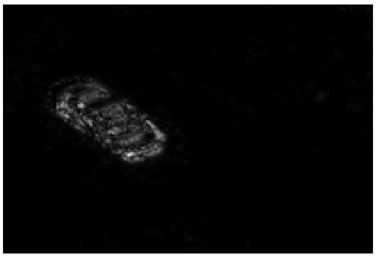
	PSNR/dB	Iteration times	Converge time
Non-preprocessing	4.48	171	5.65
Horn	24.09	68	2.47
A	26.83	48	1.66
B	25.04	31	2.03
C	24.62	22	1.9
D	24.74	6	1.34

From iterative times and convergence hours in table, we can see that enhancement is very necessary for optical flow calculation. But, the item "PNSR" is meaningless, because the final optical flow results were standardized and shown in a 256 bitmap. Algorithm ***A*** can advance accuracy and computation time, and not only improve PNSR, but also reduce the iteration times and converge time. Algorithm *D* can improve PNRS a little, and reduces both iteration times and convergence time. Algorithm *B* and *C* are similar, the calculation results of them are also almost the same, but the judgment of them are different. Therefore, their results are close. Because it filters reliable image gradient data before circulating in algorithm ***A***, diffusing velocity of moving object's border data advanced. Of course, convergence speed improved. But, we found that filtering threshold is not easy to choose in experiment. If it is not appropriate, on the contrary, real objects will lose. In future, we will do research on it furthermore, and look for a better approach. It needs a few iterative times, but iterative period is long. This method limits moving object border data to disperse to background, and avoid shortcomings about which traditional method average blindfold. It got good results. According to comparing two iterative results of σ, we choose the better after computing optical flow every time using algorithm B. In fact, it is possible to limit error when repeatedly computing, and to improve result accuracy. Based on Horn optical flow smoothness constraint hypothesis, algorithm D amends image gradient data, and choose one whose error is less. So, optical flow field improved. Obviously, this method loses meaning when acquisition hours are too large.

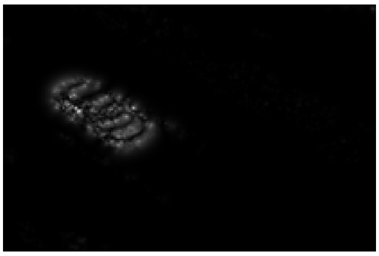
The source image is shown in Fig. 2, and there are two moving cars in the photograph.

The non-enhanced image's result is shown in Fig. 3(f), the car on the right-up corner in the source image almost disappeared. Compared to Horn method, image data diffuse more averagely. On Fig. 3(b), we can see that some more brighter white points are averagely around with some less brighter points, which means that image data diffuse enough locally. It reflects that data diffuse directly in Fig. 3(e). In motion region, white points distribute uniformly along moving direction. Principles of algorithm in Fig. 3(c) and Fig. 3(d) are similar. Comparing to Horn's method, image data diffuse more enough globally.

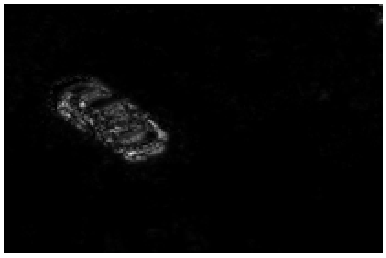
Fig. 2. Source image of a moving car.



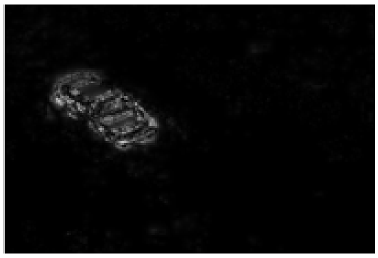
(a) Horn method



(b) Algorithm **A**



(c) Algorithm **B**



(d) Algorithm **C**

(e) Algorithm **D**

(f) The final image whose source

Fig. 3. Image processing results for different algorithm.

5. Conclusions

In this paper, we emphasize weightiness of image gradient data for optical flow field calculation based on gradient method through analyzing Horn method, and raise several approaches to improving image gradient data. From the experiment results, these approaches all improved optical flow field accuracy and computing speed with different degree, and they show validity of approaches. But, these approaches to improving optical flow calculating speed and accuracy just all confine to methods based on gradient. In fact, on one hand, Horn's global smoothness constraint has localization; on the other hand, more accurate calculation methods lay hope on new methods and advanced thought, for example, never dynamics method [9], and so on. For the future, we will try other approaches, for the sake of calculating much faster and more accurately.

References

[1]. B. K. P. Horn and B. G. Schunck, Determing optical flow, *Artificial Intelligence*, 17, 1981, pp. 185-203.

[2]. H. H. Nagel, W. Enkelmann, An Investigation of Smoothness Constraints for the Estimation of

Displacement Vector Field from Image Sequences, *IEEE Trans. Pattern Analyze and Machine Intelligence,* 1986, 8, pp. 565-593.

[3]. Lucas B., Kanade T, An interactive image registration technique with an application to stereo vision, in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981, 2, pp. 121-130.

[4]. J. Uan L., Mattthew M., Naveed S., Performance of passive ranging form image flow, *IEEE, ICIP,* 9, 2003, pp. 929 -932.

[5]. Shi Rong, and Li Zaiming, The Optical Flow Estimation with Displacement Compensation for High-speed, *IEEE, 2002*, pp. 595-599.

[6]. Yang Chunle and Shunichiro Oe, A New Gradient-based Optical Flow Method and Its Application to Motion Segmentation, *IEEE,* 2000, pp. 1225-1230.

[7]. F. Bartolini and A. Pica, Enhancement of Horn and Schunck optical flow algorithm by means of median filters, in *Proceedings of the 13th International Conference on Digital Signal Processing*, Vol. 2, 1997, pp. 503-506.

[8]. Chen Xilin, Computer Vision-Algorithm and system theory, *Tsinghua Press,* Beijing, *Guangxi Science Press,* Guilin, 1999, pp. 81-92.

[9]. Liu Guofeng, Zhu Changling, Optical flow calculation technique, *Southwest Jiaotong University Transaction,* 32, 6, 1997, pp. 656-662.