Comparison Between The Optical Flow Computational Techniques

Sri Devi Thota#1, Kanaka Sunanda Vemulapalli*2, Kartheek Chintalapati*3, Phanindra Sai Srinivas Gudipudi*4

#Associate Professor, Dept. of ECE, K L University, Andhra Pradesh, India.
**Undergraduate student, Dept. of ECE, K L University, Andhra Pradesh, India.*

***Abstract*— Optical flow is the pattern of apparent motion of objects in a visual scene caused by the relative motion between an observer and the scene. There are many methods to extract optical flow, yet there is no platform that brings out comparison on the performance of these methods. In this paper, the comparison between the results obtained by the application of two major optical flow algorithms on different sets of image sequences is brought out. Also the applications of optical flow in vehicle detection and tracking are discussed.**

***Key words*— optical flow, gradient constraint equation, aperture problem, differential technique, horn-schunck algorithm, lucas-kanade algorithm, global smoothness, spatiotemporal variations**

I. INTRODUCTION

Optical flow is defined as the apparent motion of image brightness patterns in an image sequence [11]. There are many methods to extract optical flow out of which, the gradient method is the basic method. But Gradient method cannot give a complete solution for optical flow fields because of the aperture problem. Hence to obtain a complete solution, two different Differential techniques namely Horn-Schunck algorithm and Lucas-Kanade algorithm are analysed and comparisons are made based on the results obtained.

II. GRADIENT METHOD

The basic method that has been developed to extract optical flow is the gradient method. Extraction of optical flow at the pixel level can be computed from an image sequence by making some assumptions about the spatiotemporal variations of image brightness. Our first assumption is that pixel intensities are translated from one frame to the next,

$$I(\vec{x},t) = I(\vec{x}+\vec{u},t+1)$$

where $I(\vec{x},t)$is image intensity as a function of space $\vec{x} = (x,y)^T$and time t, and $\vec{u} = (u_1,u_2)^T$is the 2D velocity. We can expand the right-hand side of the above equation in terms of first order tailor series to obtain

$$I_x.u + I_y.v + I_t = 0$$

where I_x, I_y and I_t denote the partial derivatives of intensity with respect to x, y and t respectively, $u = dx/dt$ and $v = dy/dt$denote the x and y directional components of 2 dimensional optical flow vector. The above equation is known as gradient constraint equation.

Gradient method cannot give complete solution for optical flow fields because of aperture problem i.e., normal component parallel to gradient direction can be determined but tangential component perpendicular to gradient direction remains unsolved. We have analyzed two different solutions to the aperture problem in this paper.

2.1. *Differential techniques:*

Differential techniques compute velocity from spatiotemporal derivatives of image intensity. It can also be computed from the filtered versions of the image using low-pass or band-pass filters. The first instances used first-order derivatives and are based on image translation [1], [3], [4]

$$I(X,t) = I(X - V_t, 0) \qquad (1)$$

where $V = (u,v)^T$. From the Taylor expansion of (1) or from an assumption that intensity is conserved, $dI(X,t)/dt = 0$. This implies

$$\nabla I(X,t).V + I_t(X,t) = 0 \qquad (2)$$

where $I_t(X,t)$ denote the partial derivative of $I(X,t)$ with respect to time, $\nabla I(X,t) = (I_x(X,t), I_y(X,t))^T$ and $\nabla I.V$ denotes the usual dot product. The normal component (V_n) of motion of spatial contours of constant intensity is given by (2) as $V_n = s\mathbf{n}$ where s is the normal speed and $\mathbf{n}$ is the normal direction. There are two unknown components of V in the gradient constraint equation, constrained by only one linear equation. Further constraints are therefore necessary to solve for both components of V.

Second-order differential methods use second-order derivatives to constrain 2-D velocity [4], [5], [6], [7]:

$$\begin{bmatrix} I_{xx}(X,t) & I_{yx}(X,t) \\ I_{xy}(X,t) & I_{yy}(X,t) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} I_{tx}(X,t) \\ I_{ty}(X,t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (3)$$

Equation (3) can be derived from (1) or from the conservation of$\nabla I(X,t), d\nabla I(x,t)/dt = 0$. Conservation of $\nabla I(X,t)$ implies that first-order deformations of intensity should not be present. This is therefore a stronger restriction than the gradient constraint equation on permissible motion fields. To measure image velocity, assuming $d\nabla I(x,t)/dt = 0$, the constraints in (3) may be used alone or together with (2) to yield an over-determined system of linear equations [8], [9]. However, if the aperture problem prevails in a local neighborhood, then because of the sensitivity of numerical differentiation, second-order derivatives cannot be measured accurately to determine the tangential component of *V*. As a consequence, velocity estimates from second-order methods are often assumed to be less accurate than estimates from first-order methods.

Since differential techniques cannot give the result as accurate as first order differential methods, another way to constrain $V(X)$ is to combine local estimates of component velocity and/or 2-D velocity through space and time, thereby producing more robust estimates of $V(X)$

Fig 1, Fig 2: Original images at times t, (t+Δt)

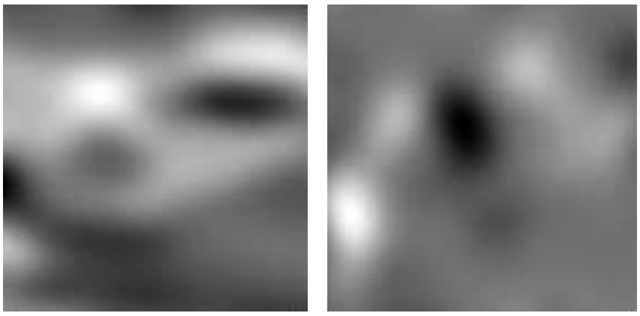

Fig 3, Fig 4: Images after applying differential technique

Fig 1 and Fig 2 are two frames of a video taken at instants *t sec* and *(t+ Δt) sec* respectively. Fig 3 and Fig 4 are the images obtained after applying differential technique to Fig 1 and Fig 2 respectively. In Fig 3, the car and the bench are highlighted with blurred shapes whereas in Fig 4, only car is highlighted .Velocity values obtained through this method are (-2.0916, -0.1200).

A. Horn-Schunck Algorithm:

One method to determine the 2D velocity using gradient constraint equation is using global smoothness constraints. In this method, the velocity field is defined in terms of the minimum of a function defined over the image.

Horn and Schunck [3] combined the gradient constraint with a global smoothness term to constrain the estimated velocity field $V(X,t) = (u(X,t), v(X,t))$, minimizing the equation

$$\int_D (\nabla I \,.\, V + I_t)^2 + \lambda^2 \left(\|\nabla u\|_2^2 + \|\nabla v\|_2^2 \right) dx \qquad (4)$$

defined over a domain D, where the magnitude of λ reflects the influence of the smoothness term. We used λ = 0.5 instead of λ = 100 as suggested by Horn and Schunck [3], because it produced better results in most of our test cases. Iterative equations in (5) are used to minimize (4) and obtain image velocity.

$$u^{n+1} = u^{-n} - \frac{I_x[I_x u^{-n} + I_y v^{-n} + I_t]}{\alpha^2 + I_x^2 + I_y^2}$$

$$v^{n+1} = v^{-n} - \frac{I_y[I_x u^{-n} + I_y v^{-n} + I_t]}{\alpha^2 + I_x^2 + I_y^2} \qquad (5)$$

where n denotes the iteration number, u^0 and v^0 denote initial velocity estimates (set to zero), and u^{-n} and v^{-n} denote neighbourhood averages of u^n and v^n. At most 100 iterations are made in all the test cases for computing the flow vectors.

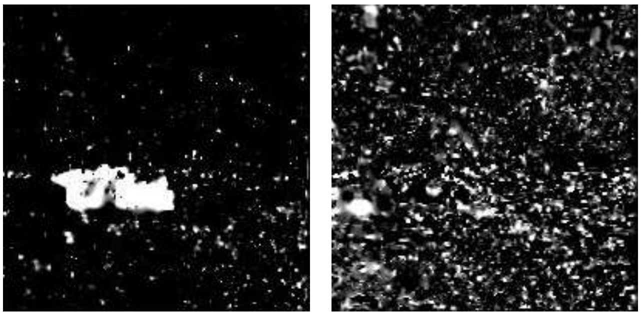

Fig 5, Fig 6: Images after applying Horn-Schunck algorithm

The original method described by Horn and Schunck used first-order differences to estimate intensity derivatives. Application of Horn-Schunck algorithm to Fig 1 and Fig 2 gives the images in Fig 5 and Fig 6. The velocities obtained through this method are (9.6837, 7.1435).

B. Lucas-Kanade Algorithm:

Though Horn-Schunck algorithm gives a complete solution for optical flow, it takes high computational time because of the iterations and hence resulting in the mathematical complexity. This can be rectified in Lucas-Kanade algorithm by implementing the concept of Least Square method. Here we find the velocity that minimizes the constraint errors. The least-squares (LS) estimator minimizes the squared errors [2] and is given as:

$$E(\vec{u}) = \sum_{\vec{x}} g(\vec{x})[\vec{u}.\nabla I(\vec{x},t) + I_t(\vec{x},t)]^2$$

where $g(\vec{x})$ is a weighting function that determines the support of the estimator. Estimator is the region within which we combine constraints. To weight constraints in the center of the neighborhood highly, giving them more influence, we consider $g(\vec{x})$to be Gaussian. The 2D velocity $\hat{u}$ is the least squares flow estimate that minimizes $E(\vec{u})$.

The minimum of $E(\vec{u})$ can be found from its critical points, where its derivatives with respect to $\vec{u}$ are zero; i.e.,

$$\frac{\partial E(u_1,u_2)}{\partial u_1} = \sum_{\vec{x}} g(\vec{x})\left[u_1 I_x^2 + u_2 I_x I_y + I_x I_t\right] = 0$$

$$\frac{\partial E(u_1,u_2)}{\partial u_2} = \sum_{\vec{x}} g(\vec{x})\left[u_2 I_y^2 + u_1 I_x I_y + I_y I_t\right] = 0$$

These equations may be rewritten in matrix form:

$$M\vec{u} = \vec{b} \qquad (6)$$

where the elements of M and $\vec{b}$ are:

$$M = \begin{bmatrix} \sum g I_x^2 & \sum g I_x I_y \\ \sum g I_x I_y & \sum g I_y^2 \end{bmatrix}, \quad \vec{b} = -\begin{pmatrix} \sum g I_x I_t \\ \sum g I_y I_t \end{pmatrix}.$$

When M has rank 2, the Least Square estimate is $\hat{u} = M^{-1}\vec{b}$. We implemented a weighted least-squares (LS) fit of local first-order constraints in equation (2) to a constant model for V in each small spatial neighborhood Ω by minimizing

$$\sum_{x\in\Omega} W^2(X)[\nabla I(X,t).V + I_t(X,t)]^2 \qquad (7)$$

where $W(X)$ denotes a window function that gives more influence to constraints at the centre of the neighbourhood than those at the periphery. The solution to (7) is given by

$$A^T W^2 A V = A^T W^2 b \qquad (8)$$

Where, for n points $X_i \in \Omega$ at a single time t,

$$A = [\nabla I(X_1), \dots, \nabla I(X_n)]^T$$
$$W = diag[W(X_1), \dots, W(X_n)]$$
$$b = -[I_t(X_1), \dots, I_t(X_n)]^T$$

The solution to (8) is $V = [A^T W^2 A]^{-1} A^T W^2 b$, which is solved in closed form when $A^T W^2 A$ is non-singular, since it is a 2x2 matrix:

$$A^T W^2 A = \begin{bmatrix} \sum W^2(X) I_x^2(X) & \sum W^2(X) I_x(X) I_y(X) \\ \sum W^2(X) I_y(X) I_x(X) & \sum W^2(X) I_x^2(X) \end{bmatrix} \quad (9)$$

where all sums are taken over points in the neighbourhood Ω.

Equations (7) and (8) may also be viewed as weighted least-squares estimates of V from estimates of normal velocities $V_n = s\boldsymbol{n}$; that is, (7) is equivalent to

$$\sum_{x\in\Omega} W^2(X) w^2(X)[V \,.\boldsymbol{n}(X) - s(X)]^2 \qquad (10)$$

where the coefficients $w^2(X)$ reflect the confidence in the normal velocity estimates; here, $w(X) = ||\nabla I(X,t)||$. The image sequence will be smoothed using spatiotemporal Gaussian filter with a standard deviation of 1.5 pixel-frames. This helps to attenuate temporal aliasing and quantization effects in the input. Fig 8 and Fig 9 are the resultant images after application of Lucas-Kanade algorithm to Fig 1 and Fig 2. The velocities obtained through this method are (4, 3).

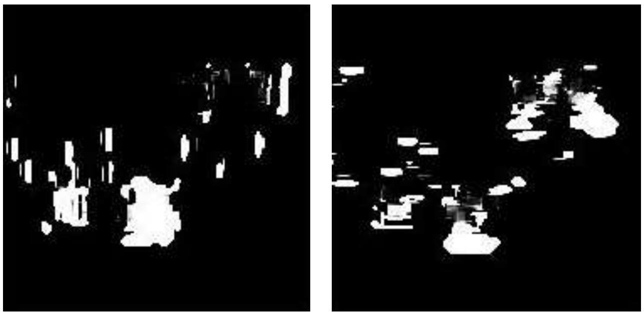

Fig 7, Fig 8: Images after applying Lucas & Kanade algorithm

2.2. Applications of Optical Flow:

Motion estimation and video compression have developed as a major aspect of optical flow research. Optical flow is being used by robotics researchers in many areas such as: object detection and tracking, image dominant plane extraction, movement detection, robot navigation and visual odometry. Optical Flow Techniques can also be used in the restoration of Non-Uniformly warped images [12]. Optical flow information has been recognized as being useful for controlling micro air vehicles.

III. RESULTS

In Differential technique, 1st order deformation should not be present to solve 2nd order equations. But this is not possible in practical cases. So using simple differential techniques will not give accurate solutions for optical field.

In Horn-Schunck algorithm, we optimize the function based on residuals from the brightness constancy. Here we improve the global smoothness but to reduce the mathematical complexity, we go for iterative method which is a time taking process. We used λ=0.5 instead of λ=100 since it produced better results in most of the cases.

In Lucas-Kanade method we use image patches and windowing methods with least squares technique.

These algorithms were tested on eight different sets of images for determining the flow field. Each algorithm produced its respective values of velocities and optical flow fields which are summarised below. Fig (a), (b) are the original images and Fig (c) is the resultant flow field.


Fig 9 (a)


Fig 9 (b)


Fig 9 (c)


Fig 10 (a)


Fig 10 (b)


Fig 10 (c)


Fig 11 (a)


Fig 11 (b)


Fig 11 (c)


Fig 12 (a)


Fig 12 (b)


Fig 12 (c)


Fig 13 (a)


Fig 13 (b)

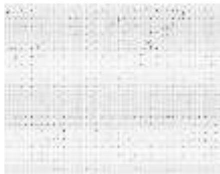
Fig 13 (c)


Fig 14 (a)


Fig 14 (b)

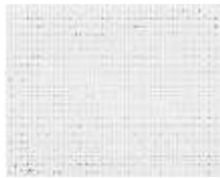
Fig 14 (c)


Fig 15 (a)


Fig 15 (b)


Fig 15 (c)


Fig 16 (a)


Fig 16 (b)

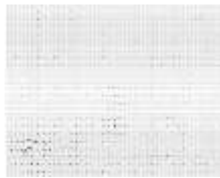
Fig 16 (c)

Table I

COMPARISON OF VELOCITIES

Method	u	v
Horn-Schunck method	9.6837	7.1435
Lucas-Kanade method	3	4

The tabulated values are obtained by the application of Horn-Schunck algorithm and Lucas-Kanade algorithm on Fig 1 and Fig 2 respectively. In Table I, **u** represents normal component parallel to gradient direction and **v** represents tangential component perpendicular to gradient direction. The velocity values are calculated in *pixels per frame*.

III. Conclusions

From Table 1 it can be inferred that, when compared to Horn-Schunck algorithm, Lucas-Kanade algorithm improves the signal strength and reduces noise giving more accurate and relatively speed results. To obtain better results in Differential Technique, we can use band pass filter to reduce the given 1st order differentiation constraints. To improve the results of Horn-Schunck algorithm, we recommend spatiotemporal presmoothing. Results of Lucas-Kanade algorithm can be improved by using FIR filters. Using FIR filters will reduce the number of delay elements.

References

[1] Fennema. C and Thompson. W, "Velocity determination in scenes containing several moving objects", in *Comput. Graph. Image Process.* 9, 1979, pp. 301-315.

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application in stereo vision" in *Seventh International Joint Conference on Artificial Intelligence*, Vancouver, 1981, pp.674–679,.

[3] Horn. B.K.R and Schunck. B.G, "Determining optical flow", in *Artificial Intelligence* Vol: 17, 1981, pp. 185-204.

[4] Nagel. H.H, "Displacement vectors derived from second-order intensity variations in image sequences", in *Comput. Graph. Image Process.* 21, 1983, pp.85-117.

[5] Tretiak. O and Pastor. L, "Velocity estimation from image sequences with second order differential operators", in *Proc. 7th Intern. Conf. Patt. Recog.*, Montreal, 1984, pp. 20-22.

[6] Nagel. H.H, "Estimation of optical flow: Relations between different approaches and some new results", in *Artificial Intelligence* 33, 1987, pp. 299-324.

[7] Uras, S., Girosi, E, Verri, A., and Ton'e, V., "A computational approach to motion perception," in *Biol. Cybern.* 60, 1988, pp.79-97.

[8] Girosi. F, Verri. A, and Torre. V, "Constraints for the computation of optical flow," *Proc. IEEE Workshop on Visual Motion*, Irvine, 1989, pp. 116-124.

[9] Tistarelli. M, Sandini. G, "Estimation of depth from motion using anthropomorphic visual sensor", in *Image Vis. Comput.* 8, 1990, pp. 271-278.

[10] John L. Barron, David J. Fleet, and Steven Beauchemin, "Performance of optical flow techniques" in *International Journal of Computer Vision* (Springer), 1994.

[11] Shahriar Negahdaripour, "Revised Definition of Optical Flow: Integration of Radiometric and Geometric Cues for Dynamic Scene Analysis", *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 20, No.9, 1998, pp. 961-979.

[12] David Clyde, Ian Scott-Fleming, Donald Fraser, Andrew Lambert, "Application of Optical Flow Techniques in the Restoration of Non-Uniformly Warped Images," in Digital Image Computing Techniques and Applications, Melbourne, Australia, 2002.

[13] David J. Fleet and Yair Weiss, "Optical Flow Estimation", in *Paragios et al. Handbook of Mathematical Models in Computer Vision*. Springer, 2006

[14] Darun Kesrarat and Vorapoj Patanavijit, "Tutorial of Motion Estimation Based on Horn-Schunk Optical Flow Algorithm in MATLAB®": AU J.T. 15(1): 8-16 (Jul 2011)