Between consecutive frames, we concern intensity **(I)** of a pixel as a function of space **(X, Y)** and time **(t)**. Therefore we can take the intensity of a pixel in the first frame as **I(x, y, t)** and when considering a displacement of **(dx, dy)** over a time dt of that same pixel in the next consecutive frame we obtain the new pixel intensity **I(x + dx, y + dy, t + dt)**.
Optical flow works on several assumptions,

1. The pixel intensities of an object do not change between 2 consecutive frames.
2. Neighboring pixels of an object also have similar motion.

Therefore, considering this constant pixel intensity assumption among consecutive frames, we can say that,

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Then by taking the Taylor series approximation of the right-hand side and removing common terms.

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx \; + \frac{\partial I}{\partial y}\, dy + \frac{\partial I}{\partial t}dt + \ldots.$$

$$I(x + dx, y + dy, t + dt) - I(x, y, t) = \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt$$

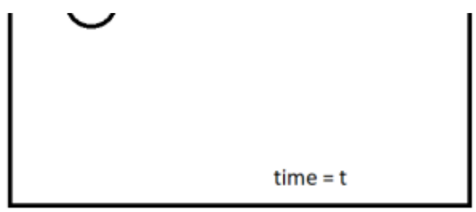


Figure 2: Position of the object at time t

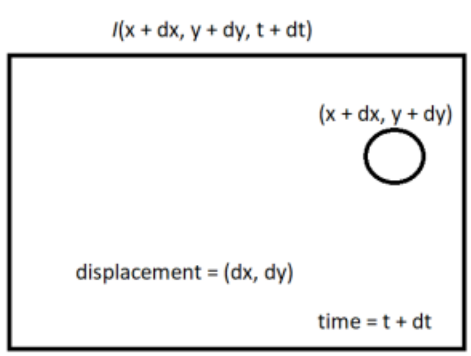


Figure 3: Position of the object after a time dt

Enhancing the Optical Flow for a Smoother Identification of Real-Time Global Motion

Optical Flow

Optical flow is a pattern of apparent motion of image objects between two consecutive frames caused by an object or camera movement. It is a 2D vector array where each vector is a displacement vector showing the movement of points from the first frame to the second. This method is called differential since it is based on local Taylor series approximations of the image signal; that is, they use partial derivatives with respect to the spatial and temporal coordinates.

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0$$

Then dividing by dt we can derive the optical flow equation

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0$$

Where

$$u = \frac{dx}{dt}, v = \frac{dy}{dt},$$

$$fx = \frac{\partial I}{\partial x}, fy = \frac{\partial I}{\partial y}, ft = \frac{\partial I}{\partial t}$$

Above equation is called Optical Flow equation. In it $\partial I/\partial x, \partial I/\partial y, \partial I/\partial t$ are the image gradients along the horizontal axis, the vertical axis and time. But *(u, v)* is unknown. We cannot solve this single equation with having two variables. So overtime several methods are provided to solve this problem

Assumptions that Optical Flow relies on

1) <u>Brightness constancy</u>: Optical flow relies on the assumption that the brightness of an object in an image sequence at pixel level remains the same over time, even in an object which moves across the frames. (Surface radiance remains fixed from one frame to the next)

2) Small motion: Optical flow assumes that the motion between two consecutive frames is very small. If the motion is too large, the optical flow algorithm may not be accurate.

3) Spatial coherence: Another assumption that optical flow relies on is the neighboring pixels having a similar motion. In other words, the motion of a pixel should be similar to the motion of the pixels around it.

4) Continuity over time: Optical flow assumes that the motion of an object is smooth over time. This means that the motion of an object should be continuous over time.

5) No occlusion: Optical flow assumes that there are no occlusions of tracking objects in the image sequence. If an object is partially or completely occluded, the algorithm may not accurately estimate its motion.

6) Constant lighting conditions: Another assumption that optical flow relies on is that image sequence remains constant over time. If there are changes in lighting, this can lead to errors in the optical flow estimation.

Paper	Research Work	Year

Distance Surface for Event-Based Optical Flow (Mohammed Almatrafi, Raymond Baldwin, Kiyoharu Aizawa)	Proposed a novel optical flow method for neuromorphic cameras (event detection cameras that report the log-intensity changes exceeding a predefined threshold at each pixel). In absence of the intensity value at each pixel location, this research introduced a notion of "distance surface"—the distance transform computed from the detected events—as a proxy for object texture. The distance surface is then used as an input to the intensity-based optical flow methods to recover the two-dimensional pixel motion. ISSUE ADDRESSED: optical flow in the presence of fast motion and occlusions.	2020
Variational Optical Flow Computation in Real Time	Investigated the usefulness of bidirectional multigrid methods for variational optical flow computations. And also showed that the efficiency of this approach even allows for real-time performance. (Variational OF involves capturing the difference between the observed OF and Predicted OF) The issue addressed: Noise-robust generalization of the Horn and Schunck technique, handled large displacements and complex motion patterns.	2005
A Database and Evaluation Methodology for Optical Flow	Proposed a new set of benchmarks and evaluation methods for the next generation of OF algorithms. To that end, they have contributed four types of data to test different aspects of OF algorithms: (1) sequences with non-rigid motion where the ground truth flow is determined by tracking fluorescent texture. (2) Realistic synthetic sequences (3) High frame-rate video used to study interpolation error. (4) Modified stereo sequences of static scenes.	2010

Optical Flow Estimation (Javier S´anchez1 , Enric Meinhardt-Llopis2 , Gabriele Facciolo)	Implementation of the OF estimation method introduced by Zach, Pock, and Bischof in 2007. This method is based on the minimization of a function containing a data term using the L 1 norm and a regularization term using the total variation of the flow. Issues addressed: Allowed discontinuities in the flow field while being more robust to noise than the classical Horn Schunk method.	2013
Gradient based optical flow estimation. (David J.Fleet, Yair Weiss)	Mentioned practical issues that some image smoothing is generally useful prior to numerical differentiation (and can be incorporated into the derivative filters). From the first-order Taylor series approximation used to derive. By smoothing the signal, one hopes to reduce the amplitudes of higher-order terms in the image and to avoid some related problems with temporal aliasing. Explained that Future research is needed to move beyond brightness constancy and smoothness. Detecting and tracking occlusion boundaries should greatly improve optical flow estimation. Similarly, prior knowledge concerning the expected form of brightness variations (e.g., given knowledge of scene geometry, lighting, or reflectance) can dramatically improve optical flow estimation. Brightness constancy is especially problematic over long image sequences where one must expect the appearance of image patches to change significantly. One promising area for future research is the joint estimation appearance and motion, with suitable dynamics for both quantities.	2005 Sept

Approach 01

Taylor Series approximation: When taking the optical flow equation by using the approximation of no intensity change, Taylor series approximation is used to remove the common terms. In this section, an investigation is intended to carry on the second-order and higher-order approximations of the Taylor Series.

Approach 02

Investigation is intended to carry out on improving the quality of the image sequences through image processing techniques and then inputting them into the optical flow.

Let I denote a pixel in a given frame, which is essentially a function of the pixel coordinates & time.

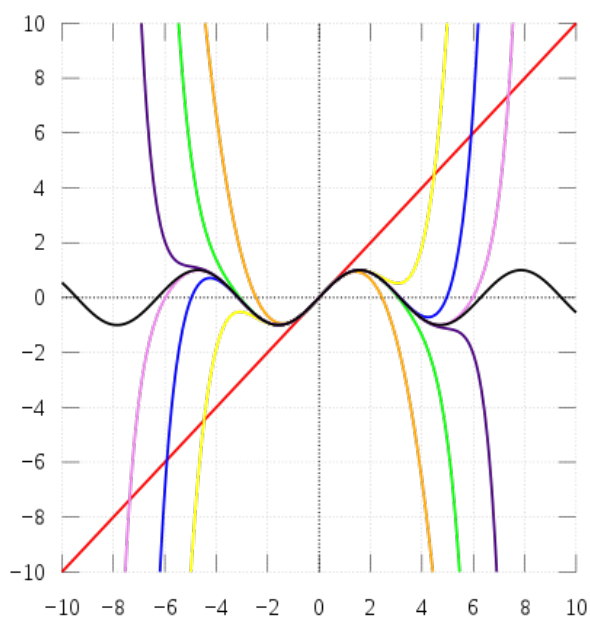
$$\therefore I \equiv I(x,\ y,\ t)$$

$$\boldsymbol{I(x,\ y,\ t) = I(x + dx,\ y + dy,\ t + dt)}$$

By the Taylor series, any function *f(x)* can be approximated by,

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(k)}(a)}{k!}(x-a)^k + \ldots$$

$$c_k = \frac{f^{(k)}(a)}{k!}$$



Now considering the Taylor series of a multivariable function,

First, consider the linear approximation for *f(x)*

$$F(\bar{x}) \equiv f(\bar{a}) + D(\bar{a})(\bar{x} - \bar{a})$$

Where D is the Jacobian of f

This is the first-order Taylor polynomial. (Linear approximation)

Let us find a quadratic approximation for *f(x)* by adding quadratic terms to the linear approximation.

For the univariable function *f(x)* the quadratic term was

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n,$$

Therefore, we need to find the second derivative of the multivariate function *f(x)*

The first derivative of at $x = a$ is the Jacobian of f at a.

For the second derivative, we take the partial derivatives of the function $Df(x)$, which is denoted by the $DDf(x)$.

This is also called the **Hessian Matrix** of f denoted by $Hf(\bar{x})$.

Therefore $Hf(x) = DDf(x)$

Hence,

$$\mathbf{H}_f = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\[2.2ex] \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\[2.2ex] \vdots & \vdots & \ddots & \vdots \\[2.2ex] \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Now, using the Hessian, the quadratic term can be written by,

$$\frac{f^{(2)}(a)}{2!}(x-a)^2 = \frac{1}{2}\left(\bar{x}-\bar{a}\right)f^{(2)}(a)(x-a) = \frac{1}{2}\left(\bar{x}-\bar{a}\right)^T Hf(\bar{a})\left(\bar{x}-\bar{a}\right)$$

Adding this to obtain second order polynomial,

$$f(\bar{x}) \approx f(\bar{a}) + Df(\bar{a})(\bar{x}-\bar{a}) + \frac{1}{2}\left(\bar{x}-\bar{a}\right)^T Hf(\bar{a})\left(\bar{x}-\bar{a}\right)$$

The second order Taylor polynomial is a better approximation for f than first order Taylor polynomial.

Now considering the scenario,

$$\bar{x} = \left(x+dx,\ y+dy,\ t+dt\right),\ \ \bar{a} = \left(x,\ y,\ t\right),\ f \equiv I$$

$$\therefore I\left(x+dx,\ y+dy,\ t+dt\right) \approx I\left(x,\ y,t\right) + Df\left(x,y,t\right)\begin{bmatrix} dx \\ dy \\ dt \end{bmatrix} + \frac{1}{2}\ \left[dx\ dy\ dt\right]\ Hf\left(x,\ y,t\right)\begin{bmatrix} dx \\ dy \\ dt \end{bmatrix}$$

Now for the Jacobian & Hessian

$$Df\left(x,\ y,t\right) = [\frac{\partial I}{\partial x}\begin{pmatrix} x \\ y \\ t \end{pmatrix}\ \frac{\partial I}{\partial y}\begin{pmatrix} x \\ y \\ t \end{pmatrix}\ \frac{\partial I}{\partial t}\begin{pmatrix} x \\ y \\ t \end{pmatrix}]$$

$$\frac{1}{2}\left[dx\ dy\ dt\right]\begin{bmatrix} \frac{\partial I^2}{\partial x_2} & \frac{\partial I}{\partial x \partial y}^2 & \frac{\partial I}{\partial x \partial t}^2 \\ \frac{\partial I}{\partial x \partial y}^2 & \frac{\partial I^2}{\partial y_2} & \frac{\partial I}{\partial y \partial t}^2 \\ \frac{\partial I}{\partial x \partial t}^2 & \frac{\partial I}{\partial y \partial t}^2 & \frac{\partial I\backslash\backslash^2}{\partial t_2} \end{bmatrix}\begin{bmatrix} dx \\ dy \\ dt \end{bmatrix}$$

$$= \frac{1}{2}[\left[\frac{\partial^2 I}{\partial x^2} dx + \frac{\partial^2 I}{\partial x \partial y} dy + \frac{\partial^2 I}{\partial x \partial t} dt\right] dx + \left[\frac{\partial^2 I}{\partial x \partial y} dx + \frac{\partial^2 I}{\partial y^2} dy + \frac{\partial^2 I}{\partial y \partial t} dt\right] dy + \left[\frac{\partial^2 I}{\partial x \partial t} dx + \frac{\partial^2 I}{\partial y \partial t} dy + \frac{\partial^2 I}{\partial t^2} dt\right] dt]$$

$$= \frac{\partial^2 I}{\partial x \partial y}\left(dxdy\right) + \frac{\partial^2 I}{\partial x \partial t}\left(dxdt\right) + \frac{\partial^2 I}{\partial y \partial t}\left(dydt\right) + \frac{1}{2}[\left[\frac{\partial^2 I}{\partial x^2}(\partial x)^2 + \frac{\partial^2 I}{\partial y^2}(\partial y)^2 + \frac{\partial^2 I}{\partial t^2}(\partial t)^2\right]$$

By applying back in above approximation,

$$\therefore I(x+dx,\ y+dy,\ t+dt) \approx I(x,\ y,t) + \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} + \frac{\partial^2 I}{\partial x \partial y}\left(dxdy\right) + \frac{\partial^2 I}{\partial x \partial t}\left(dxdt\right) + \frac{\partial^2 I}{\partial y \partial t}\left(dydt\right) + \frac{1}{2}[\left[\frac{\partial^2 I}{\partial x^2}(\partial x)^2 + \frac{\partial^2 I}{\partial y^2}(\partial y)^2 + \frac{\partial^2 I}{\partial t^2}(\partial t)^2\right]$$

$$0 \approx + \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} + \frac{\partial^2 I}{\partial x \partial y}\left(dxdy\right) + \frac{\partial^2 I}{\partial x \partial t}\left(dxdt\right) + \frac{\partial^2 I}{\partial y \partial t}\left(dydt\right) + \frac{1}{2}[\left[\frac{\partial^2 I}{\partial x^2}(\partial x)^2 + \frac{\partial^2 I}{\partial y^2}(\partial y)^2 + \frac{\partial^2 I}{\partial t^2}(\partial t)^2\right]$$

$$0 = \left(\frac{\partial^2 I}{\partial x \partial y} + \frac{1}{2}\frac{\partial^2 I}{\partial x^2} + \frac{\partial I}{\partial x}\right)dx + \left(\frac{\partial^2 I}{\partial y \partial t} + \frac{1}{2}\frac{\partial^2 I}{\partial y^2} + \frac{\partial I}{\partial y}\right)dy + \left(\frac{\partial^2 I}{\partial x \partial t} + \frac{1}{2}\frac{\partial^2 I}{\partial t^2} + \frac{\partial I}{\partial t}\right)dt$$

Differentiate w.r.t. t;

$$0 = \underbrace{\left(\frac{\partial^2 I}{\partial x \partial y} + \frac{1}{2}\frac{\partial^2 I}{\partial x^2} + \frac{\partial I}{\partial x}\right)}_{I_x} dx/dt + \underbrace{\left(\frac{\partial^2 I}{\partial y \partial t} + \frac{1}{2}\frac{\partial^2 I}{\partial y^2} + \frac{\partial I}{\partial y}\right)}_{I_y} dy/dt + \underbrace{\left(\frac{\partial^2 I}{\partial x \partial t} + \frac{1}{2}\frac{\partial^2 I}{\partial t^2} + \frac{\partial I}{\partial t}\right)}_{I_t}$$

So, from this we have finally derived our New Optical Flow equation.

In this we can determine I_x and I_y change in the pixel position.

Similarly, I_t is the gradient along time which we can calculate.

However, $\frac{dx}{dt}\ \frac{dy}{dt}$ are unknown.

Hence B becomes an equation with 2 unknown variables, We set $x_t = \dfrac{dx}{dt} \quad y_t = \dfrac{dy}{dt}$

Hence B becomes,

$$I_x x_t \ + \ I_y y_t + I_t = 0$$

We attempt to solve this by several methods, first we'll focus on Lucas-Kanade method.

Lucas-Kenade Method

We have assumed that all neighboring pixels have the same motion.

In here, the neighborhood definition becomes ambiguous, which we define to be a 3×3 patch around the point in the case.

$\therefore$ all 9 points in the patch have the same motion.

We determine I_x , I_y &,I_t for these pts

We are now left with 9 equations with 2 unknown variables, which are over determined.

Hence, we use the least square fit method we approximate the system of 9 equations

Consider the points in the selected path to be $q_1, q_2, \ldots\ldots., q_9$

Then,

$$I_x(q_1) + \ I_y(q_1) = - \ I_t(q_1)$$
$$I_x(q_2) + \ I_y(q_2) = - \ I_t(q_2)$$
$$.$$
$$.$$
$$.$$
$$I_x(q_9) + \ I_y(q_9) = - \ I_t(q_9)$$

Representing in a matrix,

$$\begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_9) & I_y(q_9) \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} = (-) \begin{bmatrix} I_t(q_1) \\ I_t(q_2) \\ I_t(q_9) \end{bmatrix}$$

A V b

We consider x_t & y_t are parameters to be learnt to determine the best fit line by the variables $I_x(q)$ & $I_y(q)$ by applying examples / data points q1 through q9

We use ordinary least square method for the purpose, where,

$$A^T A v = A^T b$$

$$\therefore v = (A^T A)^{-1} A^T b$$

Clarification: consider a matrix A as follows,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \quad -> \quad A^T A = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

$$\therefore A^T A = \begin{bmatrix} a_{11}{}^2 + a_{21}{}^2 + a_{31}{}^2 & a_{11}a_{12} + a_{21}a_{22} + a_{31}a_{32} \\ a_{12}a_{11} + a_{22}a_{21} + a_{32}a_{31} & a_{12}{}^2 + a_{22}{}^2 + a_{32}{}^2 \end{bmatrix}$$

$$\therefore A^T A = \begin{bmatrix} \sum_{i=1}^{3} a_{i1}{}^2 & \sum_{i=1}^{3} a_{i1}a_{i2} \\ \sum_{i=1}^{3} a_{i2}a_{i1} & \sum_{i=1}^{3} a_{i2}{}^2 \end{bmatrix}$$

When applied with the image pixels I def.

$$A^T A = \begin{bmatrix} \sum_{i=1}^{n} I_x(q_i)^2 & \sum_{i=1}^{n} I_x(q_i) I_y(q_i) \\ \sum_{i=1}^{n} I_x(q_i) I_y(q_i) & \sum_{i=1}^{n} I_y(q_i)^2 \end{bmatrix}$$

$A^T A$ matrix is often called the structure tensor of the image at the center point p of the patch selected., Also

$$A^T b = \begin{bmatrix} -\sum_{i=1}^{n} I_x(q_i) I_t(q_i) \\ -\sum_{i=1}^{n} I_y(q_i) I_t(q_i) \end{bmatrix}$$

Now using these we obtain, for our example

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{9} I_x(q_i)^2 & \sum_{i=1}^{9} I_x(q_i) I_y(q_i) \\ \sum_{i=1}^{9} I_x(q_i) I_y(q_i) & \sum_{i=1}^{n} I_y(q_i)^2 \end{bmatrix} \begin{bmatrix} -\sum_{i=1}^{9} I_x(q_i) I_t(q_i) \\ -\sum_{i=1}^{9} I_y(q_i) I_t(q_i) \end{bmatrix}$$

By using the solutions, we determine the complete solution to the problem.

i.e, we know of x_t & y_t that solves, hence solving the optical flow equations.

We obtain optical flows along the 2- dimensions of the image.