Robust Optical Flow Computation: A Higher-Order Differential Approach

Chanuka Algama*, Kasun Amarasinghe†
University of Kelaniya
Carnegie Mellon University
* chanukaravishan25@gmail.com, † kamarasi@andrew.cmu.edu

***Abstract*—In the domain of computer vision, optical flow stands as a cornerstone for unraveling dynamic visual scenes. However, the challenge of accurately estimating optical flow under conditions of large displacement remains an open question. The image flow constraint is vulnerable to substantial nonlinear elements, and rapid temporal or spatial changes. Inaccurate approximations inherent in numerical differentiation techniques can further amplify such intricacies. In response, this research proposes an innovative algorithm for optical flow computation, utilizing the higher precision of second-order Taylor series approximation within the differential estimation framework. By embracing this mathematical underpinning, the research seeks to extract more information about the behavior of the function under complex real-world scenarios and estimate the motion of areas with a lack of texture. An impressive showcase of the algorithm's capabilities emerges through its performance on renowned optical flow benchmarks such as KITTI (2015) and Middlebury. The average endpoint error (AEE), which computes the Euclidian distance between the calculated flow field and the ground truth flow field, stands notably diminished, validating the effectiveness of the algorithm in handling complex motion patterns. This further indicates its potential for practical application in a range of real-world scenarios that require accurate global motion estimations, such as autonomous navigation, video surveillance, flight stabilization in drones, video stabilization, and motion-based recognition, where accurate motion estimation between consecutive frames is imperative.**

***Index Terms*—Optical Flow, Motion estimation, Taylor-series Approximation, Dense Optical Flow**

I. Introduction

Optical flow has been a fundamental tool in the field of computer vision for decades. It plays a pivotal role in understanding motion dynamics within visual scenes by estimating the pattern of apparent motion of image objects between two successive image frames caused by the motion of an object or camera. It is a 2D vector array where each vector is a displacement vector representing the movement between two subsequent frames. [1]. Applications range from autonomous navigation [2] [3] [4] [5], breast tumor detection [6], bladder cancer detection [7], drone/video stabilization, traffic monitoring [8] and fluid mechanics [9] etc. Accurate optical flow estimation is imperative for these tasks, as it allows systems to track movement, recognize patterns, and interact intelligently with dynamic environments.

Despite its broad application, reliable computation of flow fields under complex real-world scenarios is a major issue in Optical flow. Particularly when dealing with, large displacements, rapid temporal or spatial changes, and regions lacking texture often result in inaccurate flow computations. Over the past two decades, numerous techniques and novel concepts have evolved to overcome this. In particular, the past decade has seen extraordinary improvement with the development of advanced computer vision benchmarks such as KITTI [10], Middlebury [11], and MPI-Sintel [12] which provided the algorithms significant novel hurdles in real-world settings.

Existing computation methods are largely based on differential approach which involves using spatial and temporal derivatives of the image intensities to compute the flow. while block matching methods, energy-based methods, correlation based methods and learning based methods are other prominent optical flow estimation approaches. Most of these approaches, struggle in capturing nonlinear motion patterns and rapid spatial transformations. The image flow constraint, while fundamental, tends to break down under large nonlinear deformations, leading to significant errors.One key source of inaccuracy lies in the reliance on numerical differentiation techniques, which often amplify minimal errors when estimating spatial and temporal derivatives. These are particularly pronounced in real-world scenarios characterized by fast-moving objects, occlusions, and complex environments. Moreover, many optical flow algorithms fail to handle regions with minimal texture effectively, as there are no strong visual features to guide the motion estimation process. As a result, there remains a need for more sophisticated approaches that can account for these intricacies while maintaining computational efficiency.

In response, this research proposes an innovative algorithm for the computation within the differential estimation framework utilizing the higher precision of the second-order Taylor series approximation. By embracing this mathematical underpinning the research seeks to extract more information about the behavior of the function under under complex real-world scenarios and estimate the motion of areas with lack of texture. We evaluate the performance of our algorithm using widely recognized benchmarks, KITTI (2015) and Middlebury, which test its ability to handle complex motion patterns in real-world scenarios. Results demonstrate a notable reduction in Average Endpoint Error (AEE), validating the effectiveness of the method in estimating motion with high precision.

II. Related work

In this section, we discuss flow estimation methods closely related to this work and their limitations.

A. Differential Approaches

The seminal work by Horn and Schunck [13] introduced the first global differential method for optical flow estimation, formulating the problem as an energy minimization task. Their approach assumes constant brightness intensity across frames, which leads to the following derivation:

Let $I(x, y, t)$ be the image intensity function at position (x, y) and time t. Assuming constant intensity across two consecutive frames, we have:

$$I(x, y, t) = I(x + u, y + v, t + dt)$$

Where u and v represent the horizontal and vertical components of the optical flow velocity, and dt is the time difference between frames.

Using Taylor series expansion around (x, y, t), we approximate the right-hand side of the equation:

$$I(x+u, y+v, t+dt) = I(x, y, t) + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + dt\frac{\partial I}{\partial t} \\ + \text{higher-order terms}$$

Subtracting $I(x, y, t)$ from both sides and ignoring higher-order terms gives us:

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

Dividing by dt, we obtain the fundamental optical flow equation:

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

Where $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$ denote the spatial and temporal derivatives of the image intensity function $I(x, y, t)$.

Horn-Schunck applies smoothness constraints to regularize the flow field. Lucas and Kanade [14] developed a more localized differential approach, focusing on small patches of an image to estimate flow based on local gradient information. While computationally efficient and suitable for small displacements, this method suffers in the presence of rapid changes in motion or in low-texture regions where gradients are weak. Both approaches remain limited by their reliance on linear approximations, which cannot capture complex, nonlinear motion patterns.

B. Feature-Based and Matching Methods

Feature-based methods attempt to track prominent features (such as corners or edges) across frames to estimate optical flow. Shi and Tomasi [15] extended Lucas-Kanade's method by improving feature selection, ensuring that only well-defined features are tracked. However, these methods are highly dependent on the quality and density of the features, which often limits their application in areas with insufficient texture or weak features.

Block matching and correlation-based methods, divide the image into blocks and attempt to match corresponding blocks between frames. While these methods can handle large displacements better than differential approaches, they assume the motion of all the pixels inside a block having a constant motion.

C. Energy Minimization and Variational Methods

To address the issue of large displacements and the need for smooth flow fields, energy minimization techniques such as those in Brox et al. [16] extend the Horn and Schunck model by introducing robust penalty functions and handling larger displacements. These methods use coarse-to-fine warping strategies and integrate nonlinear optimization to achieve better accuracy. However, their dependence on handcrafted energy functions makes them sensitive to parameter tuning and less adaptable to highly dynamic environments.

D. Learning-Based Approaches

The emergence of deep learning has significantly advanced optical flow estimation. FlowNet [17] was one of the first deep neural networks designed to directly estimate optical flow from raw image pairs. Its follow-up, FlowNet 2.0 [18], used a combination of convolutional networks and coarse-to-fine refinements to enhance performance, especially in large displacement scenarios. Similarly, PWC-Net [19], EpciFlow [20] and RAFT [21] have demonstrated the power of end-to-end learning in optical flow estimation, achieving state-of-the-art performance on various benchmarks.

III. Overview

While these deep learning models achieve impressive results, they require large training datasets and considerable computational resources, limiting their applicability in resource-constrained environments. Moreover, the black-box nature of these models makes it difficult to interpret their behavior in complex scenarios, and they can struggle in situations not well-represented in the training data, such as low-texture regions or extreme lighting conditions.

Despite significant progress, among existing non-learning based methods, algorithms still face several limitations in handling large displacements, nonlinearity, and textureless regions. This study acknowledges the assumptions made in past studies that higher-order terms in Taylor series approximation can be neglected considering an arbitrarily small time step $\delta(t)$ between consecutive frames, and also, the motion of a pixel between consecutive frames is linear. However, this is often violated in real-world scenarios, as the time steps are typically fixed [22]. By highlighting this practical limitation, this research recognizes the need to investigate the impact of this assumption. As described by Shang Hong-Lai and Baba Vemuri [23], neglecting higher-order terms in optical flow computation can lead to a temporal aliasing problem, resulting in inaccuracies in the computed flow fields. However, most works that utilized higher-order approximations have stopped short of fully leveraging their potential, either due to the computational burden or the difficulty in balancing approximation accuracy with model simplicity. To address this, the study proposes a novel algorithm that leverages second-order Taylor series approximation and perturbation theory to avoid such assumptions and limitations.

Additionally, previous studies have often relied on the Barron dataset [24] to evaluate their Optical flow algorithms. However, the dataset primarily consists of simple transformations and is characterized by a relatively high frame rate of 45-60 frames per second (fps). While it has provided valuable insights over the years, its limitations in assumptions-based algorithms on laboratory setting scenarios with lower frame rates have been recognized.

To address this, the study will employ the KITTI benchmark for evaluation [10] which is a widely accepted benchmark among computer vision community, developed by Karlsruhe Institute of Technology and Toyota Institute of Technology in Chicago [25]. The dataset offers a more realistic representation of driving scenes captured at a standard frame rate of 24 fps, which most often resembles in movies and TV. Furthermore, the dataset comprises stereo camera setups, which provide a vision similar to human binocular vision with additional depth information. By utilizing the KITTI dataset, the research aims to evaluate the performance of the proposed optical flow algorithm under more realistic conditions that closely resemble real-world driving conditions, which is crucial for developing reliable algorithms suitable for autonomous driving systems and other related applications.

IV. PROPOSED ALGORITHM

In this section, we present the mathematical derivation of the proposed algorithm, a comparative analysis of the existing and proposed algorithm, and the benchmark suites utilized for testing and evaluation.

A. *Mathematical Derivation*

The foundation of optical flow estimation relies on the assumption of brightness consistency. Mathematically this approximation can be expressed as,

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

We linearize the expression by applying Taylor series expansion around (x, y, t), it allows us to approximate the intensity function $I(x + dx, y + dy, t + dt)$ in terms of its partial derivatives with respect to spatial coordinates and time. The Taylor series is an extremely powerful tool for estimating functions that are too challenging to compute otherwise, using it any function *f(x)* can be approximated by,

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

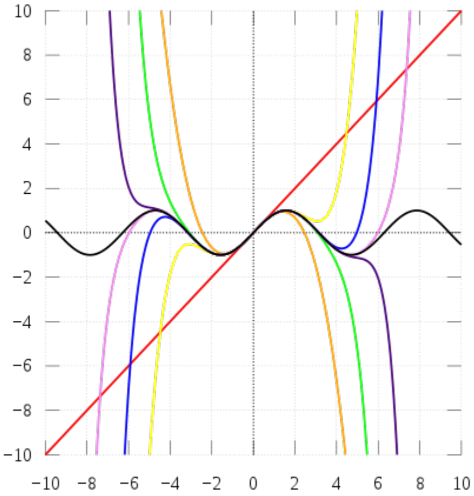

Fig. 1: Taylor Polynomial Approximations of sin x at x = 0

The accuracy of a Taylor polynomial improves as its degree increases, allowing it to approximate the actual function [26] more effectively. Figure 1 illustrates this relationship between the function $sin(x)$ and its Taylor approximations using polynomials of degrees 1, 3, 5, 7, 9, 11, and 13, specifically around $x = 0$.

Extending this idea to the Taylor series approximation of a multivariable function, we begin by examining the linear approximation for the function $f(x)$. This can be expressed as:

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

Here, D represents the Jacobian matrix of the function f, evaluated at point $\overline{a}$. This formulation is the basis for the first-order Taylor polynomial, which provides a linear approximation of the function around point $\overline{a}$.

To improve upon this approximation, quadratic terms are introduced to derive a more accurate second-order Taylor polynomial. For a univariate function $f(x)$, the Taylor series is given by:

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

where $f^{(n)}(a)$ represents the n-th derivative of the function at a. This series expands upon the linear approximation, adding terms for higher degrees of accuracy as the degree of the polynomial increases.

Subsequently, The partial derivatives of the function $Df(x)$, when taken again to compute the second-order partial derivatives, result in the Hessian Matrix of the function f. This matrix, which provides a compact representation of all second-order partial derivatives, is denoted as $Hf(x)$ and is defined as follows:

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

accordingly,

$$Hf(x) = D^2f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

[1]

By using the Hessian matrix, the quadratic term can be expressed in the following manner:

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

Subsequently, by including the quadratic terms derived from the Hessian matrix, the approximation can be extended as:

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

In light of the provided scenario for the derivation,

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

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

Considering the combined effects of the Jacobian and Hessian matrices,

$$0 = Df(x,y,t) = \left[\frac{\partial I}{\partial x}\begin{pmatrix}x\\y\\t\end{pmatrix}, \frac{\partial I}{\partial y}\begin{pmatrix}x\\y\\t\end{pmatrix}, \frac{\partial I}{\partial t}\begin{pmatrix}x\\y\\t\end{pmatrix}\right] + \frac{1}{2}\left[dx\,dy\,dt\right]\begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial x \partial t} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} & \frac{\partial^2 I}{\partial y \partial t} \\ \frac{\partial^2 I}{\partial x \partial t} & \frac{\partial^2 I}{\partial y \partial t} & \frac{\partial^2 I}{\partial t^2} \end{bmatrix}\begin{bmatrix}dx\\dy\\dt\end{bmatrix}$$

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

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

By incorporating the derived results back into the approximation,

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

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

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

Differentiating with respect to t,

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

Finally, as a result of a comprehensive analysis and a meticulous mathematical procedure, the novel optical flow equation has been successfully derived.

In this,

$$Ix = (\frac{\partial^2 I}{\partial x \partial y} + \frac{1}{2}\frac{\partial^2 I}{\partial x^2} + \frac{\partial I}{\partial x})$$

$$Iy = (\frac{\partial^2 I}{\partial y \partial t} + \frac{1}{2}\frac{\partial^2 I}{\partial y^2} + \frac{\partial I}{\partial y})$$

are the spatial gradients along horizontal and vertical axis respectively. Similarly,

$$It = (\frac{\partial^2 I}{\partial x \partial t} + \frac{1}{2}\frac{\partial^2 I}{\partial t^2} + \frac{\partial I}{\partial t})$$

is the intensity gradient over time. To determine the motion components in the x and y directions, denoted by $u = dt/dx$, $v = dt/dy$, Lucas -kanade method can be utilized [14].

[1] The Hessian matrix is a square matrix whose dimension is equal to the number of variables of the function. For example, for a function with 3 variables, the Hessian matrix will be a 3x3 dimensional matrix

B. Comaparative Analysis of the Existing and Proposed Algorithm

The proposed algorithm introduces several significant modifications compared to the existing algorithm. One key modification is the integration of a more advanced motion model that takes into account not only local linear pixel intensity changes but also complex non-linear changes through higher-order motion patterns and temporal dependencies. Moreover, the proposed algorithm incorporates advanced regularization techniques to promote smoothness in the estimated flow fields. By incorporating spatial and temporal constraints, the algorithm reduces noise and outliers by utilizing GaussianBlur, which is a widely used image filtering technique that helps to reduce noise and detail the image while preserving the overall structure and edges in the image [20]. Furthermore, The algorithm consists of 2 input parameters alpha and delta, which are the smoothing parameter, and the parameter for the tolerance, respectively. which gives the ability to manually adjust the characteristics of the scene conditions of the input data.

C. Benchmark Suite for Testing and Evaluation

In this section, we outlines the selected benchmark suits selected such as, KITTI and Middlebury, and the evaluation metrics employed in the research.

1) KITTI Vision Benchmark suite: The KITTI dataset comprises a collection of real-world driving scenes captured from a moving vehicle [10]. The dataset encompasses a variety of challenging environmental conditions such as lighting, weather, and scene variations. By incorporating the KITTI dataset, the algorithm's performance can be assessed under realistic and challenging scenarios, providing insights into its robustness and adaptability.

2) Middlebury Dataset: The Middlebury dataset is a renowned benchmark suite designed for optical flow evaluation. Over the years, many novel optical flow methods have evaluated their performance using the Middlebury dataset. It encompasses a set of image pairs with manually annotated ground truth optical flow. The algorithm's accuracy and robustness can be evaluated across different motion types by utilizing the Middlebury dataset, enabling a comprehensive analysis of its performance.

3) Evaluation Metrics:

- Average Endpoint Error (AEE): This is considered a standard evaluation metric for evaluating optical flow algorithms. It calculates the Euclidean distance between the estimated flow fields and the ground truth flow fields. A lower AEE indicates higher accuracy in flow estimation.
- Percentage of Erroneous Pixels (PEP): The PEP metric measures the percentage of pixels where the endpoint error exceeds a predefined threshold.

V. Experimental Validations

We compared the performance of the proposed algorithm against state-of-the-art optical flow algorithms in 3 distinct scenarios. Then we evaluate the proposed method on the KITTI vision benchmark suite, which contains 200 videos of driving scenes in different lighting conditions. We quantitatively evaluate the precision of the estimated flow fields of the algorithm with novel optical flow constraint equation vs existing constraint equation using Average-End-Point error. We further utilized the Middlebury dataset to examine the performance of the algorithm.

A. Comaprison Against Existing Algorithms

We begin by comparing the performance of our algorithm with the optical flow implementation provided by OpenCV, a widely respected library within the computer vision community known for its robust and reliable algorithm implementations. The comparison is conducted across three distinct scenes. Since these tests lack ground truth flow fields, we assess the visualized flow fields from both algorithms to estimate the amount of global motion each captures.

Figure 2 illustrates the comparison between the proposed algorithm and the existing algorithm. In tracking global motion, The red arrows indicate the direction of the estimated motion while the length of the arrows indicates the magnitude of the flow fields. Across the three test scenarios, the proposed algorithm has clearly captured more global motion

Figure 4.2 comparison demonstrates that the proposed algorithm has identified more global motion of the different objects in the image. This particular experiment unintentionally has broken an underline assumption in optical flow that there should be enough features or texture in the scenario to detect any motion.

The sphere motion comparison in Figure 4.3 demonstrates that **the proposed algorithm has accurately captured the motion of the upper part of the sphere. At the same time, the existing algorithm has failed to capture any movement in the upper part of the ball.**

The conducted multiple sets of rigorous comparisons over challenging scenarios demonstrate the proposed algorithms' superior performance over understanding global motion. It exhibits notable advantages over existing methods in accuracy and robustness. The algorithm captures fine details, preserves object boundaries, and generates smoother motion estimations.

References

[1] D. Sun, S. Roth, J. P. Lewis, and M. J. Black, "Learning Optical Flow," en, in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds., vol. 5304, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 83–97, ISBN: 978-3-540-88689-1 978-3-540-88690-7. DOI: 10.1007/978-3-540-88690-7_7. [Online]. Available: http://link.springer.com/10.1007/978-3-540-88690-7_7 (visited on 12/03/2022).

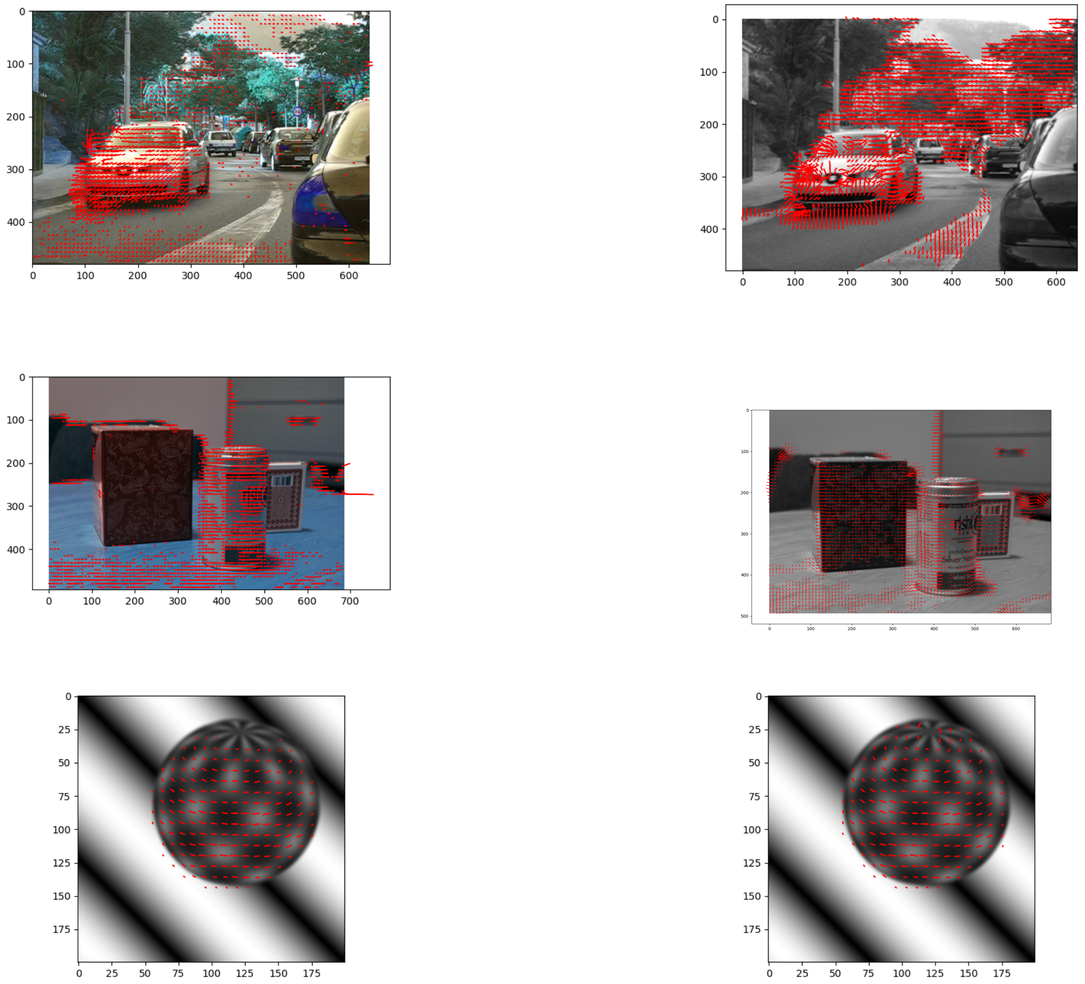

Fig. 2: Comparison of flow fields from the existing and proposed algorithms across three distinct test scenes.

Existing Algorithm (left) **Proposed Algorithm** (right)

[2] H. Chao, Y. Gu, and M. Napolitano, "A Survey of Optical Flow Techniques for Robotics Navigation Applications," en, *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 361–372, Jan. 2014, ISSN: 1573-0409. DOI: 10.1007/s10846-013-9923-6. [Online]. Available: https://doi.org/10.1007/s10846-013-9923-6 (visited on 05/26/2023).

[3] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," en, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 3061–3070, ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298925. [Online]. Available: http://ieeexplore.ieee.org/document/7298925/ (visited on 05/26/2023).

[4] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," en, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 3061–3070, ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298925. [Online]. Available: http://ieeexplore.ieee.org/document/7298925/ (visited on 05/26/2023).

[5] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[6] M. Abdel-Nasser, A. Moreno, H. A. Rashwan, and D. Puig, "Analyzing the evolution of breast tumors through flow fields and strain tensors," en, *Pattern Recognition Letters*, Pattern Recognition Techniques in Data Mining, vol. 93, pp. 162–171, Jul. 2017, ISSN: 0167-8655. DOI: 10.1016/j.patrec.2016.11.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786551630318X (visited on 05/26/2023).

[7] T. Weibel, C. Daul, D. Wolf, R. Rösch, and F. Guillemin, "Graph based construction of textured large field of view mosaics for bladder cancer diagnosis," en, *Pattern Recognition*, vol. 45, no. 12, pp. 4138–

4150, Dec. 2012, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2012.05.023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320312002695 (visited on 05/26/2023).

[8] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," en, *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, Apr. 2003, ISSN: 0262-8856. DOI: 10.1016/S0262-8856(03)00004-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885603000040 (visited on 05/26/2023).

[9] Y. Kameda, A. Imiya, and T. Sakai, "Hierarchical properties of multi-resolution optical flow computation," vol. 7584, Oct. 2012, pp. 576–585, ISBN: 978-3-642-33867-0. DOI: 10.1007/978-3-642-33868-7_57.

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, ISSN: 1063-6919, Jun. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[11] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," en, *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, Mar. 2011, ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-010-0390-2. [Online]. Available: http://link.springer.com/10.1007/s11263-010-0390-2 (visited on 12/03/2022).

[12] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A Naturalistic Open Source Movie for Optical Flow Evaluation," en, in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2012, pp. 611–625, ISBN: 978-3-642-33783-3. DOI: 10.1007/978-3-642-33783-3_44.

[13] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[14] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International journal of computer vision*, vol. 61, no. 3, pp. 211–231, 2005.

[15] M. Bansal, M. Kumar, M. Kumar, and K. Kumar, "An efficient technique for object recognition using shi-tomasi corner detection algorithm," *Soft Computing*, vol. 25, no. 6, pp. 4423–4432, 2021.

[16] T. Brox, "Optical flow: Traditional approaches," in Oct. 2021, pp. 921–925, ISBN: 978-3-030-63415-5. DOI: 10.1007/978-3-030-63416-2_600.

[17] P. Fischer, A. Dosovitskiy, E. Ilg, *et al.*, "Flownet: Learning optical flow with convolutional networks," Apr. 2015.

[18] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[19] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," Jun. 2018, pp. 8934–8943. DOI: 10.1109/CVPR.2018.00931.

[20] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," en, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 1164–1172, ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298720. [Online]. Available: http://ieeexplore.ieee.org/document/7298720/ (visited on 05/26/2023).

[21] A. Ranftl, F. Alonso-Fernandez, S. Karlsson, and J. Bigun, "Real-time AdaBoost cascade face tracker based on likelihood map and optical flow," *IET Biometrics*, vol. 6, no. 6, pp. 468–477, Nov. 2017, arXiv:2210.13885 [cs], ISSN: 2047-4938, 2047-4946. DOI: 10.1049/iet-bmt.2016.0202. [Online]. Available: http://arxiv.org/abs/2210.13885 (visited on 05/26/2023).

[22] T. Brox and J. Malik, "Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation," en, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, Mar. 2011, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.143. [Online]. Available: http://ieeexplore.ieee.org/document/5551149/ (visited on 05/26/2023).

[23] S.-H. Lai and B. C. Vemuri, "[No title found]," en, *International Journal of Computer Vision*, vol. 29, no. 2, pp. 87–105, 1998, ISSN: 09205691. DOI: 10.1023/A:1008005509994. [Online]. Available: http://link.springer.com/10.1023/A:1008005509994 (visited on 12/03/2022).

[24] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," en, *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–466, Sep. 1995, ISSN: 0360-0300, 1557-7341. DOI: 10.1145/212094.212141. [Online]. Available: https://dl.acm.org/doi/10.1145/212094.212141 (visited on 12/03/2022).

[25] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[26] J. Fosso-Tande, "Applications of taylor series," Aug. 2013.