ReynoldsFlow: Physics-Inspired Spatiotemporal Flow Representation for Video Understanding

Yu-Hsi Chen[1], Ching-Kai Lin[2], PingKong Huang[2], and Chin-Tien Wu[2,✉]

[1] The University of Melbourne, Parkville, Australia
[2] Department of Applied Mathematics, National Yang Ming Chiao Tung University, Hsinchu, 300, Taiwan ctw@math.nctu.edu.tw

Abstract. Video understanding has largely relied on deep spatiotemporal architectures, including 3D convolutional networks and optical flow (OF) based models. While effective, these methods are often computationally expensive and depend on heuristic motion representations that are sensitive to illumination, scale, and structural changes. To address these limitations, we propose ReynoldsFlow, a physics-inspired representation grounded in the Reynolds transport theorem (RTT) and Helmholtz-Hodge decomposition (HHD). ReynoldsFlow decomposes motion into curl-free (CF) and divergence-free (DF) components, providing a principled and interpretable characterization of scene dynamics. By coupling intensity information with decomposed motion cues, it produces dynamics-aware, texture-preserving features that boost downstream tasks such as pose estimation, action recognition, and tiny object detection. Lightweight and modular, ReynoldsFlow can be readily integrated into existing architectures. Experiments across diverse benchmarks show that ReynoldsFlow consistently matches or surpasses existing approaches, offering improved generalizability and computational efficiency.

Keywords: Video Understanding · Physics-Inspired Vision · Reynolds Transport Theorem · Helmholtz-Hodge Decomposition

1 Introduction

Computer vision is increasingly integrated into daily life, with video functionality central to modern smartphones. Beyond casual use, video understanding supports a wide range of applications, including stabilization, interpolation, object detection [25, 67–69], multi-object tracking [1, 9, 10], and pose estimation [7, 16, 38, 56]. Progress has been largely driven by deep learning (DL) architectures such as RNNs [12, 66], LSTMs [17, 65], and, more recently, transformer- and Mamba-based spatiotemporal models [15, 21, 29, 37, 54]. Despite their strong empirical performance, these methods remain computationally expensive, rely on carefully tuned heuristics, and offer limited interpretability, as their representations are rarely grounded in physical motion principles.

✉ Corresponding author.

To address these challenges, we propose ReynoldsFlow, a physics-inspired video representation. Combining OF [49], RTT [59], and the HHD [2], ReynoldsFlow produces interpretable spatiotemporal features in a training-free, unsupervised manner. By deriving video representations directly from data rather than from heuristic motion representations, the resulting features are more efficient and generalizable. Our main contributions are summarized as follows:

1. **ReynoldsFlow formulation.** We propose ReynoldsFlow, an unsupervised motion representation that decomposes motion into CF and DF components via RTT and HHD, providing physically interpretable dynamics.
2. **ReynoldsFlow representation.** We provide a visualization that integrates decomposed motion magnitudes with image appearance cues, yielding a more interpretable and robust representation than HSV-based visualization.
3. **Comprehensive evaluation.** We validate ReynoldsFlow on multiple video benchmarks, showing its effectiveness for video representation learning.

2 Related Work

2.1 Optical Flow and Video Motion Estimation

Early OF methods established the foundational tradeoff between local accuracy and global smoothness, pioneered by the works of Lucas-Kanade (LK) [31] and Horn-Schunck (HS) [20]. Subsequent classical approaches focused on overcoming the fragility of these formulations. Variational methods and polynomial expansion algorithms [6, 14] significantly improved sub-pixel precision, while others prioritized robustness against illumination changes and noise [52, 64]. To handle large displacements and complex occlusions, later heuristic techniques integrated sparse feature matching with dense interpolation [27, 58, 61]. While interpretable and mathematically grounded, these techniques can be less effective when handling complex real-world motion under computational constraints.

The paradigm subsequently shifted toward data-driven architectures. CNN-based frameworks demonstrated the viability of end-to-end prediction [11], though they initially lacked fine-grained detail. This limitation drove a wave of architectural innovations, including cascaded refinement [22], cost-volume pyramids [42, 50], and edge-aware sparse matching [43]. This evolution culminated in models such as RAFT [53], which set a new standard by iteratively refining dense flows using learned contextual correlations. Recent extensions have further pushed the boundaries of temporal consistency and cross-domain generalization [34, 35, 57]. However, despite their strong performance, deep OF networks typically require large-scale training data and task-specific optimization.

Application-driven integrations of OF highlight these practical limitations. For UAV detection, [51] embeds an LK-inspired layer into YOLO, while [32] leverages Farneback OF for golf pose estimation using handcrafted features. Both demonstrate ingenuity but remain constrained by the limited expressiveness of flow visualizations. To our knowledge, no existing work effectively leverages mathematically rigorous, physics-aware OF images as direct network inputs to substantially improve downstream detection or action recognition tasks.

2.2 Decomposition-based Representations

The HHD provides a principled mathematical framework for separating a velocity field into CF, DF, and harmonic components [5, 46]. This formulation has been widely applied in computer vision and graphics for interpreting flow fields and solving boundary value problems. Early work demonstrated applications of discrete HHD to image processing, enabling structural analysis of visual data [36]. To enhance robustness, convex optimization formulations were later introduced to regularize image flows and stabilize the decomposition process [63]. Beyond motion analysis, decomposition-based formulations have found broad applications in image representation, enabling structural organization of large-scale image databases [19], signal separation through sparse decomposition [13], and multimodal image fusion via decomposition and sparse coding [70].

More recently, decomposition-based techniques have been extended to practical vision tasks, such as illumination compensation and texture enhancement in imaging [60]. Physics-inspired neural architectures further push this line of research, as in HDNet [40], which leverages HHD to design interpretable deep networks for flow estimation. Collectively, these works show that decomposition-based representations can improve interpretability and robustness in visual analysis. However, most approaches remain confined to spatial formulations, focusing on single-frame decomposition of images or velocity fields while neglecting temporal dynamics. This lack of spatiotemporal integration limits their applicability to modern video understanding tasks, where temporal coherence is critical.

2.3 Spatiotemporal Modules in Video Learning

Modeling temporal dynamics is fundamental to video understanding, as motion provides critical cues for object behavior, scene evolution, and activity recognition. To capture such dynamics, DL methods incorporated 3D convolutions and recurrent architectures to learn spatiotemporal representations directly from video sequences [39, 41, 55, 62]. However, their reliance on large-scale supervised training and computationally intensive optimization often limits efficiency and generalization in data-scarce settings. Physics-inspired approaches offer a principled alternative to purely data-driven temporal modeling. Previous work employed the RTT to treat video sequences as continuous frame flows, enabling theoretically grounded analysis of temporal changes [47]. Building on this idea, ReynoldsFlow combines RTT with the HHD to decompose motion into two flow components as an unsupervised approach. This method bridges principled physical modeling with modern feature learning, producing generalizable representations without task-specific training.

3 ReynoldsFlow

In this section, we formulate ReynoldsFlow based on RTT and HHD, establishing a physics-inspired framework for velocity field estimation. We further present the visualization scheme and analyze its computational efficiency.

3.1 Preliminaries

Helmholtz-Hodge decomposition Let $\boldsymbol{v} \in C^1(\Omega, \mathbb{R}^2)$ be a continuously differentiable velocity field defined on a domain Ω. According to the HHD theorem [2], $\boldsymbol{v}$ can be uniquely decomposed into the sum of a CF (irrotational) component $\boldsymbol{v}_c$ and a DF (solenoidal) component $\boldsymbol{v}_d$:

$$\boldsymbol{v} = \boldsymbol{v}_c + \boldsymbol{v}_d, \tag{1}$$

where $\boldsymbol{v}_c$ satisfies $\nabla \times \boldsymbol{v}_c = 0$ and $\boldsymbol{v}_d$ satisfies $\nabla \cdot \boldsymbol{v}_d = 0$.

Reynolds transport theorem Consider a smooth scalar function $f = f(\boldsymbol{p}, t)$ defined on a time-dependent domain $\Omega(t)$. A grayscale video is represented as $f(\boldsymbol{p}(t^n), t^n)$, where f denotes the intensity at pixel location $\boldsymbol{p}(t^n)$ in the domain $\Omega(t^n)$, and t^n is the timestamp of the n-th frame. The RTT [59] states that:

$$\begin{aligned} \frac{d}{dt}\int_{\Omega(t)} f\, dA &= \int_{\Omega(t)} \frac{\partial f}{\partial t}\, dA + \int_{\partial\Omega(t)} f(\boldsymbol{v}\cdot\boldsymbol{n})\, dS \\ &= \int_{\Omega(t)} \frac{\partial f}{\partial t}\, dA + \int_{\Omega(t)} \nabla\cdot(f\boldsymbol{v})\, dA \\ &= \int_{\Omega(t)} \left(\frac{\partial f}{\partial t} + \nabla f\cdot\boldsymbol{v} + f\nabla\cdot\boldsymbol{v}\right) dA. \end{aligned} \tag{2}$$

where $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{p}, t)$ is the velocity field at $\boldsymbol{p} \in \Omega(t)$ and time t. For video footage, if the brightness in the region $\int_{\Omega(t)} f dA$ is constant and the velocity field $\boldsymbol{v}$ is DF (i.e., $\boldsymbol{v} = \boldsymbol{v}_d$) for all time, then Eq. (2) reduces to:

$$0 = \int_{\Omega(t)} \left(\frac{\partial f}{\partial t} + \nabla f\cdot\boldsymbol{v}_d\right) dA. \tag{3}$$

Assuming the integrand in Eq. (3) vanishes pointwise, and $\boldsymbol{v}_d$ is piecewise constant, Eq. (3) reduces to the traditional OF.

Area Jacobian in domain transformation Consider a region $\Omega(t)$ evolving over time t under the influence of a velocity field $\boldsymbol{v}$. Let $\boldsymbol{p}^n = \begin{pmatrix} x^n \\ y^n \end{pmatrix} \in \Omega^n = \Omega(t^n)$ at time t^n, with corresponding velocity field $\boldsymbol{v}^n = \begin{pmatrix} v_x^n \\ v_y^n \end{pmatrix}$, and assume a uniform time increment $\Delta t = t^{n+1} - t^n$. Using the explicit Euler method, the domain transformation from Ω^n to Ω^{n+1} can be approximated by:

$$\boldsymbol{p}^{n+1} \approx \boldsymbol{p}^n + \boldsymbol{v}^n(\boldsymbol{p}^n)\Delta t. \tag{4}$$

From Eq. (4), the differential wedge product $dx \wedge dy$ at $\boldsymbol{p}^{n+1}$ can be written as:

$$\begin{aligned} dx^{n+1} \wedge dy^{n+1} &\approx (dx^n + dv_x^n \Delta t) \wedge \left(dy^n + dv_y^n \Delta t\right) \\ &= dx^n \wedge dy^n + dx^n \wedge dv_y^n \Delta t + dv_x^n \Delta t \wedge dy^n + dv_x^n \Delta t \wedge dv_y^n \Delta t. \end{aligned}$$

Using the fact that

$$dv_x = \frac{\partial v_x}{\partial x}dx + \frac{\partial v_x}{\partial y}dy, \quad dv_y = \frac{\partial v_y}{\partial x}dx + \frac{\partial v_y}{\partial y}dy,$$

we obtain

$$dx^{n+1} \wedge dy^{n+1} = dx^n \wedge dy^n + \nabla \cdot \boldsymbol{v}^n \, dx^n \wedge dy^n \, \Delta t + \mathcal{O}((\Delta t)^2).$$

For sufficiently small Δt, this simplifies to:

$$dx^{n+1} \wedge dy^{n+1} \approx (1 + \nabla \cdot \boldsymbol{v}^n \Delta t) \, dx^n \wedge dy^n. \tag{5}$$

As a result, $(1 + \nabla \cdot \boldsymbol{v}^n \Delta t)$ approximates the Jacobian determinant relating area elements dA^n in Ω^n to dA^{n+1} in Ω^{n+1}. In other words,

$$dA^{n+1} \approx (1 + \nabla \cdot \boldsymbol{v}^n \Delta t) \, dA^n. \tag{6}$$

3.2 CF Component: Irrotational Field

To generalize beyond the brightness constancy and DF assumptions, we invoke the RTT on each local patch $\omega(t)$ combined with the HHD Eq. (1) to rewrite:

$$\frac{d}{dt}\int_{\omega(t)} f \, dA = \int_{\omega(t)} \left(\frac{\partial f}{\partial t} + \nabla f \cdot \boldsymbol{v}_d + \nabla f \cdot \boldsymbol{v}_c + f \nabla \cdot \boldsymbol{v}_c \right) dA. \tag{7}$$

Applying the explicit Euler method, the LHS of Eq. (7) is approximated as:

$$\begin{aligned} \text{LHS} &= \frac{1}{\Delta t}\left(\int_{\omega^{n+1}} f^{n+1} \, dA^{n+1} - \int_{\omega^n} f^n \, dA^n \right), \\ &\approx \frac{1}{\Delta t}\int_{\omega^n} \left[(1 + \nabla \cdot \boldsymbol{v}^n \Delta t) f^{n+1} - f^n \right] dA^n, \quad \text{by Eq. (6)} \\ &= \int_{\omega^n} \left(\frac{f^{n+1} - f^n}{\Delta t} + f^{n+1} \nabla \cdot \boldsymbol{v}^n \right) dA^n. \end{aligned} \tag{8}$$

Next, using the Taylor approximation:

$$f^{n+1} - f^n \approx \frac{\partial f^n}{\partial t}\Delta t + \nabla f^n \cdot (\boldsymbol{v}^n \Delta t) = \frac{\partial f^n}{\partial t}\Delta t + \nabla f^n \cdot \begin{pmatrix} \Delta x^n \\ \Delta y^n \end{pmatrix},$$

and the HHD of the velocity field $\boldsymbol{v}$,

$$\begin{aligned} \text{LHS} &\approx \int_{\omega^n} \left(\frac{\partial f^n}{\partial t} + \nabla f^n \cdot \boldsymbol{v}^n + f^{n+1} \nabla \cdot \boldsymbol{v}^n \right) dA^n, \\ &= \int_{\omega^n} \left(\frac{\partial f^n}{\partial t} + \nabla f^n \cdot (\boldsymbol{v}_c^n + \boldsymbol{v}_d^n) + f^{n+1} \nabla \cdot \boldsymbol{v}_c^n \right) dA^n. \end{aligned}$$

Similarly, the RHS of Eq. (7) is

$$\text{RHS} = \int_{\omega^n} \left(\frac{\partial f^n}{\partial t} + \nabla f^n \cdot (\boldsymbol{v}_c^n + \boldsymbol{v}_d^n) + f^n \nabla \cdot \boldsymbol{v}_c^n \right) dA^n, \tag{9}$$

Equating both sides and defining $\delta f^n = f^{n+1} - f^n$ gives

$$\int_{\omega^n} \delta f^n \nabla \cdot \boldsymbol{v}_c^n \, dA^n = 0.$$

Integration by parts leads to the variational form:

$$\int_{\partial\omega^n} \delta f^n \boldsymbol{v}_c^n \cdot \boldsymbol{n} \, dS^n - \int_{\omega^n} \nabla \delta f^n \cdot \boldsymbol{v}_c^n \, dA^n = 0. \tag{10}$$

We can compute the velocity field $\boldsymbol{v}_c^n$ from Eq. (10) by assuming it remains constant within each local window patch ω^n. Specifically, we compute $\boldsymbol{v}_c^n$ on a 3×3 window patch, denoted as $\omega_{3\times3}^n$. To approximate the boundary integral term in Eq. (10), we apply Simpson's rule:

$$\int_{\partial\omega_{3\times3}^n} \delta f^n \boldsymbol{v}_c^n \cdot \boldsymbol{n} \, dS^n \approx [(\delta f_b^n)_x, (\delta f_b^n)_y] \cdot \boldsymbol{v}_c^n, \tag{11}$$

where

$$(\delta f_b^n)_x = \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -4 & 0 & 4 \\ -1 & 0 & 1 \end{bmatrix} * \delta f^n \text{ and } (\delta f_b^n)_y = \frac{1}{3} \begin{bmatrix} 1 & 4 & 1 \\ 0 & 0 & 0 \\ -1 & -4 & -1 \end{bmatrix} * \delta f^n.$$

The domain integral term in Eq. (10) is approximated as:

$$\int_{\omega_{3\times3}^n} \nabla \delta f^n \cdot \boldsymbol{v}_c^n \, dA^n = \int_{\omega^n} [(\nabla \delta f^n)_x, (\nabla \delta f^n)_y] \cdot \boldsymbol{v}_c^n \, dA^n \approx [(\nabla \delta f_\omega^n)_x, (\nabla \delta f_\omega^n)_y] \cdot \boldsymbol{v}_c^n, \tag{12}$$

where

$$(\nabla \delta f_\omega^n)_x = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * \left(\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \delta f^n \right) \text{ and } (\nabla \delta f_\omega^n)_y = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * \left(\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \delta f^n \right).$$

By substituting the discrete approximations Eq. (11) and Eq. (12) into the variational form Eq. (10), we derive the irrotational velocity field $\boldsymbol{v}_c^n$. To ensure spatial smoothness and robustness against local noise, we apply a Gaussian smoothing kernel G, defining the CF velocity field as:

$$\boldsymbol{v}_c^n = \begin{bmatrix} G * ((\delta f_b^n)_x - (\nabla \delta f_\omega^n)_x) \\ G * ((\delta f_b^n)_y - (\nabla \delta f_\omega^n)_y) \end{bmatrix}. \tag{13}$$

This irrotational velocity field $\boldsymbol{v}_c^n$ isolates the residual expansion fluxes, non-rigid deformations, and varying illuminations. We can then plug $\boldsymbol{v}_c^n$ back into the transport equation to resolve the complete motion field.

3.3 DF Component: Solenoidal Field

To resolve the DF component $\boldsymbol{v}_d^n$, we equate the discrete formulation in Eq. (8) with the continuous formulation in Eq. (9):

$$\int_{\omega^n} \left(\frac{f^{n+1}-f^n}{\Delta t} + f^{n+1}\nabla \cdot \boldsymbol{v}_c^n \right) dA^n = \int_{\omega^n} \left(\frac{\partial f^n}{\partial t} + \nabla f^n \cdot (\boldsymbol{v}_c^n + \boldsymbol{v}_d^n) + f^n \nabla \cdot \boldsymbol{v}_c^n \right) dA^n. \tag{14}$$

The discrete temporal difference can be related to the continuous derivative using the Taylor expansion

$$f^{n+1} = f^n + \frac{\partial f^n}{\partial t}\Delta t + \frac{\partial^2 f^n}{\partial t^2}\frac{(\Delta t)^2}{2} + \mathcal{O}((\Delta t)^3).$$

Ignoring higher-order terms gives

$$\frac{f^{n+1} - f^n}{\Delta t} - \frac{\partial f^n}{\partial t} \approx \frac{\partial^2 f^n}{\partial t^2}\frac{\Delta t}{2}.$$

Substituting this relation into Eq. (14) and isolating the advection term yields

$$\begin{aligned} \int_{\omega^n} \nabla f^n \cdot \boldsymbol{v}_d^n \, dA^n &\approx \int_{\omega^n} \left((f^{n+1} - f^n)\nabla \cdot \boldsymbol{v}_c^n - \nabla f^n \cdot \boldsymbol{v}_c^n + \frac{\partial^2 f^n}{\partial t^2}\frac{\Delta t}{2} \right) dA^n \\ &\approx \int_{\omega^n} \left(\delta f^n \nabla \cdot \boldsymbol{v}_c^n - \nabla f^n \cdot \boldsymbol{v}_c^n + \frac{f^{n+1} - 2f^n + f^{n-1}}{2\Delta t} \right) dA^n, \end{aligned} \tag{15}$$

where the second-order temporal derivative is approximated using central differencing. Although Eq. (15) captures intensity acceleration, the second-order term is noise-sensitive and vanishes under constant-velocity translation. We therefore impose the assumption of local intensity preservation, where the total intensity within the moving patch remains approximately constant:

$$\int_{\omega^n} f^{n+1} dA^n \approx \int_{\omega^n} f^n dA^n.$$

In the absence of geometric divergence, $\boldsymbol{v}_c^n = 0$, the discrete LHS of Eq. (14) becomes zero. Therefore, physical consistency requires:

$$\int_{\omega^n} \left(\frac{\partial f^n}{\partial t} + \nabla f^n \cdot (\boldsymbol{v}_c^n + \boldsymbol{v}_d^n) + f^n \nabla \cdot \boldsymbol{v}_c^n \right) dA^n = 0. \tag{16}$$

By the localization theorem, the integrand vanishes pointwise. Approximating $\frac{\partial f^n}{\partial t} \approx \delta f^n$ removes the acceleration term while preserving first-order consistency. Rearranging gives the divergence-compensated residual map D:

$$\nabla f^n \cdot \boldsymbol{v}_d^n = \underbrace{-\delta f^n}_{\text{Temporal Diff.}} - \underbrace{\nabla f^n \cdot \boldsymbol{v}_c^n}_{\text{CF Advection}} - \underbrace{f^n \nabla \cdot \boldsymbol{v}_c^n}_{\text{CF Dilation}} \equiv D. \tag{17}$$

When $\boldsymbol{v}_c^n = 0$, Eq. (17) naturally reduces to the classical brightness constancy constraint $\nabla f^n \cdot \boldsymbol{v}_d^n = -\delta f^n$, which corresponds to Eq. (3).

To estimate the DF velocity field $\boldsymbol{v}_d^n = (u_d, v_d)^\top$, we assume it is locally constant within a neighborhood ω. Following the LK framework [4,31], we solve a Gaussian-weighted least squares problem:

$$\min_{\boldsymbol{v}_d^n} \sum_{\boldsymbol{x}\in\omega} W(\boldsymbol{x}) \left(\nabla f^n(\boldsymbol{x}) \cdot \boldsymbol{v}_d^n - D(\boldsymbol{x})\right)^2, \tag{18}$$

where $W(\boldsymbol{x})$ is a Gaussian weighting function. This yields

$$\begin{bmatrix} \sum W (f_x^n)^2 & \sum W f_x^n f_y^n \\ \sum W f_x^n f_y^n & \sum W (f_y^n)^2 \end{bmatrix} \begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} \sum W f_x^n D \\ \sum W f_y^n D \end{bmatrix}, \tag{19}$$

where f_x^n and f_y^n denote spatial gradients. Finally, the complete ReynoldsFlow is formulated as $\boldsymbol{v}_R^n = \boldsymbol{v}_c^n + \boldsymbol{v}_d^n$.

3.4 ReynoldsFlow Representation

Flow is commonly visualized in the HSV color space, where motion magnitude and direction are encoded as hue and saturation [3]. However, the nonlinear HSV-to-RGB transformation can introduce perceptual inconsistencies, particularly in low-texture regions, complex illumination, or dynamic motion scenarios. These limitations may degrade the reliability of flow representations for downstream tasks such as tiny object detection. To address this issue, we provide an alternative visualization that augments flow with additional appearance cues. Specifically, we construct a three-channel representation

$$\boldsymbol{F}_{\mathrm{R}}^n = [|\boldsymbol{v}_d^n|,\ |\boldsymbol{v}_c^n|,\ f^n],$$

where the red and green channels encode the magnitudes of the DF and CF, respectively, while the blue channel preserves the current frame intensity f^n. By integrating motion magnitudes with image appearance, this representation improves velocity field interpretability and robustness to visual ambiguity, particularly for tiny or fast-moving objects. As shown in Fig. 1, $\boldsymbol{F}_{\mathrm{R}}^n$ produces sharper and more stable features across diverse datasets.

4 Experimental Results

We first evaluate ReynoldsFlow against classical and DL-based OF methods under four controlled scenarios: (1) geometric divergence, (2) illumination variation, (3) horizontal translation, and (4) pure rotation. We then assess the ReynoldsFlow representation on downstream vision tasks, including pose estimation on GolfDB [33], action recognition on HMDB51 [28] and UCF101 [48], and object detection on Anti-UAV [24], ARD100 [18], and UAVDB [8].

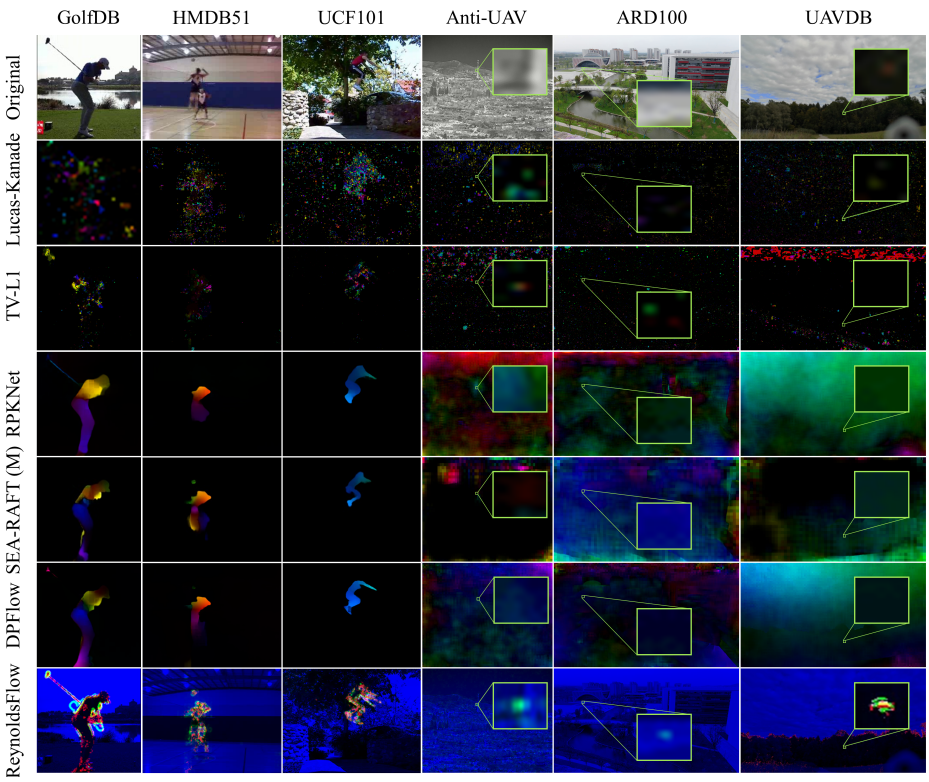

Fig. 1: Top row: Sample frames from six benchmark datasets. Middle rows: HSV visualizations from five OF methods. Bottom row: ReynoldsFlow representations.

We compare ReynoldsFlow with 13 input representations: (1) RGB, (2) grayscale, classical OF methods including (3) Horn-Schunck [20], (4) Lucas-Kanade [31], (5) Farneback [14], (6) Brox [6], (7) TV-L1 [64], (8) DeepFlow [58], (9) PCAFlow [61], and (10) DIS [27], as well as DL-based methods (11) RPKNet [35], (12) SEA-RAFT [57], and (13) DPFlow [34]. Classical methods were chosen for their training-free nature, consistent with ReynoldsFlow, while DL-based methods used official pretrained models due to the absence of ground-truth velocity fields in most datasets. All velocity fields are visualized in HSV following [30], where hue and saturation encode motion direction and magnitude, respectively. ReynoldsFlow follows the visualization scheme in Sec. 3.4.

4.1 Controlled Evaluation of Decomposed Motion Fields

Setup A Sony DSC-QX100 camera is mounted on a motorized linear slider (GVM 1.5D-120) supported by a tripod, with a 13×12 checkerboard calibration board containing 18 mm squares serving as the reference target. Camera cali-

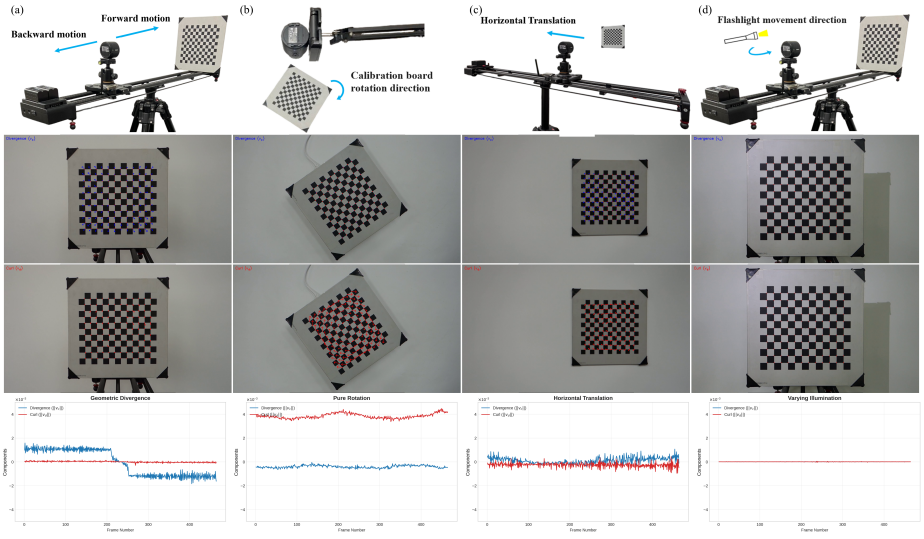

Fig. 2: Top row: Experimental setups for the controlled scenarios. Second and third rows: $\boldsymbol{v}_c$ and $\boldsymbol{v}_d$ velocity fields with blue and red arrows. Bottom row: Temporal mean magnitudes of the CF and DF components for each scenario.

bration is performed using the MATLAB R2025b Camera Calibration Toolbox. Linear slider is performed at a constant speed of 2.81 cm/s, while the checkerboard is mounted on a motorized turntable rotating clockwise at a constant angular velocity of 0.197 rad/s. Camera calibration and optical alignment are summarized in Tab. 1, which reports the calibrated intrinsic parameters, lens distortion coefficients, reprojection error, and center-alignment statistics. Optical alignment is quantified by the horizontal and vertical offsets (dx, dy) between the checkerboard center and the optical axis, together with the radial deviation $r = \sqrt{dx^2 + dy^2}$, reported as the mean $\pm$ standard deviation over ten repeated trials. Four controlled scenarios are evaluated, as shown in Fig. 2. In (a), *Geometric Divergence*, the camera moves along its optical axis toward or away from the checkerboard, producing zoom-in and zoom-out effects over a camera-checkerboard distance ranging from 58.7 to 82.8 cm. In (b), *Pure Rotation*, the camera remains fixed while the checkerboard rotates at a camera-checkerboard distance of 72.3 cm. In (c), *Horizontal Translation*, the camera moves leftward along the slide rail while the checkerboard remains stationary, with the camera-checkerboard distance varying from 88.7 to 93.2 cm at a vertical separation of 85.5 cm. In (d), *Varying Illumination*, the camera-checkerboard distance remains fixed at 57.4 cm, while a light source follows a semicircular trajectory above the camera with a light-checkerboard distance ranging from 59.0 to 69.3 cm.

Results Let $\boldsymbol{v}_i$ and $\hat{\boldsymbol{v}}_i$ denote the ground-truth and estimated flow vectors at checkerboard corner $i \in \{1, \ldots, N\}$, and $\boldsymbol{v}_t$ and $\hat{\boldsymbol{v}}_t$ represent the vectors across

Table 1: Verification of the experimental setup. The left partition details the calibrated camera intrinsics, while the right partition reports the center alignment residual offsets.

Camera Intrinsic Parameters					Center Alignment Verification				
Resolution	f_x (px)	f_y (px)	c_x (px)	c_y (px)	Case	dx (px)	dy (px)	r (px)	$r_{\max}$ (px)
1920×1080	2667.4945	2666.5458	965.9914	537.9983	Geometric Div.	0.0 ± 0.7	-0.6 ± 0.5	1.0 ± 0.4	2.3
Repr. Error	k_1	k_2	p_1	p_2	Pure Rotation	0.9 ± 0.4	-7.8 ± 0.5	7.9 ± 0.5	8.9
0.0863 (px)	0.0530	-0.7239	-1.85×10^{-4}	4.90×10^{-4}	Varying Illum.	1.3 ± 0.1	-5.2 ± 0.1	5.3 ± 0.1	5.6

valid frames $t \in \{1, \ldots, T\}$. Spatial accuracy is measured by the standard Average Endpoint Error (AEPE) and Average Angular Error (AAE) [3], while temporal expansion consistency and directional alignment are captured by Sparse Radial Correlation (R_{sp}) and Angular Cosine Similarity (ACS). Let m_t and $\hat{m}_t$ denote the ground-truth and estimated radial magnitudes, with temporal means $\bar{m}$ and $\bar{\hat{m}}$, respectively. These four metrics are formulated as:

$$R_{sp} = \frac{\sum_{t=1}^{T}(\hat{m}_t - \bar{\hat{m}})(m_t - \bar{m})}{\sqrt{\sum_{t=1}^{T}(\hat{m}_t - \bar{\hat{m}})^2 \sum_{t=1}^{T}(m_t - \bar{m})^2}}, \qquad \text{ACS} = \frac{1}{T}\sum_{t=1}^{T} \frac{\hat{\boldsymbol{v}}_t \cdot \boldsymbol{v}_t}{\|\hat{\boldsymbol{v}}_t\|_2 \|\boldsymbol{v}_t\|_2},$$

$$\text{AEPE} = \frac{1}{T}\sum_{t=1}^{T} \|\hat{\boldsymbol{v}}_t - \boldsymbol{v}_t\|_2, \;\; \text{AAE} = \frac{1}{T}\sum_{t=1}^{T} \arccos\left(\frac{\hat{\boldsymbol{v}}_t \cdot \boldsymbol{v}_t + 1}{\sqrt{(\|\hat{\boldsymbol{v}}_t\|_2^2 + 1)(\|\boldsymbol{v}_t\|_2^2 + 1)}}\right).$$

To validate the theoretical zero-divergence and zero-curl constraints, we report repeated controlled experiments in Tab. 2. As shown in the right panel of Fig. 2, *Geometric Divergence* captures expansion-to-contraction transitions with near-zero curl, while *Pure Rotation* exhibits strong curl with minimal divergence. *Varying Illumination* and *Horizontal Translation* produce near-zero values for both components, demonstrating robustness to rigid motion and photometric variations. Notably, the *Horizontal Translation* sequence also includes slight divergence-related variation caused by perspective scaling as the checkerboard moves toward and away from the camera.

4.2 Pose Estimation on GolfDB

We evaluate pose estimation on GolfDB [33], which contains 1,400 videos at 160× 160 resolution with subjects occupying nearly the full frame. Each golf swing is annotated with eight events: Address (A), Toe-up (TU), Mid-backswing (MB), Top (T), Mid-downswing (MD), Impact (I), Mid-follow-through (MFT), and Finish (F), captured from face-on and down-the-line views. We use SwingNet [33] with a setup consisting of sequence length 64, 10 frozen layers, batch size 22, 2,000 iterations, six workers, initializing with the pre-trained MobileNetV2 [44] and fine-tuning on GolfDB. Performance is measured using the Percentage of Correct Events (PCE) with four-fold cross-validation. For real-time 30 fps videos, we use a tolerance of $\delta = 1$; for slow-motion, $\delta = \max(\lfloor \frac{N}{f} \rceil, 1)$, where N is the number of frames from Address to Impact and f the sampling frequency. As shown in Tab. 3, our input achieves the highest PCE of 0.812.

Table 2: Quantitative comparison of OF and ReynoldsFlow velocity fields across four controlled tests. Results are reported as mean ± standard deviation over 10 runs. Best results are in **bold**, second best are underlined.

Algorithms	Geometric Divergence				Varying Illumination			
	R_{sp} ↑	ACS ↑	AEPE ↓	AAE (°) ↓	R_{sp} ↑	ACS ↑	AEPE ↓	AAE (°) ↓
Horn-Schunck [20]	0.9491±0.08	0.9695±0.01	0.1885±0.04	10.3510±1.02	0.6251±0.12	**0.9999**±0.00	0.0026±0.00	0.1385±0.05
Lucas-Kanade [31]	0.8812±0.18	0.9895±0.01	0.1052±0.02	5.4210±0.48	0.1245±0.02	0.9988±0.00	0.0105±0.00	0.4512±0.06
Farneback [14]	0.9452±0.11	0.9965±0.00	0.0635±0.01	3.3412±0.29	0.6110±0.15	**0.9999**±0.00	0.0022±0.00	0.1221±0.04
Brox [6]	0.9421±0.14	0.9938±0.01	0.0832±0.02	4.3051±0.28	0.3698±0.15	**0.9999**±0.00	0.0041±0.00	0.2125±0.05
TV-L1 [64]	0.8781±0.19	0.9945±0.01	0.0805±0.02	4.1522±0.42	0.2791±0.08	0.9997±0.00	0.0052±0.00	0.2810±0.07
DeepFlow [58]	0.8410±0.21	0.9928±0.01	0.0915±0.02	4.7120±0.51	-0.1685±0.06	0.9994±0.00	0.0195±0.00	1.0821±0.06
PCAFlow [61]	0.8175±0.24	0.9721±0.01	0.2054±0.02	11.2810±0.52	-0.0521±0.02	0.9905±0.01	0.0892±0.03	4.8510±1.25
DIS [27]	0.9315±0.13	0.9951±0.00	0.0751±0.02	3.9850±0.38	0.3045±0.04	**0.9999**±0.00	0.0047±0.00	0.2525±0.06
RPKNet [35]	0.7812±0.26	0.9855±0.01	0.1465±0.02	7.7120±0.51	-0.1345±0.05	0.9981±0.00	0.0502±0.01	2.7650±0.42
SEA-RAFT (M) [57]	0.7625±0.25	0.9825±0.01	0.1652±0.02	9.1250±0.68	-0.0712±0.01	0.9991±0.00	0.0258±0.01	1.4150±0.28
DPFlow [34]	0.8715±0.17	0.9835±0.01	0.1551±0.02	8.2015±0.52	-0.1575±0.08	0.9978±0.00	0.0581±0.02	3.2350±0.23
ReynoldsFlow (Ours)	**0.9654**±0.06	**0.9975**±0.00	**0.0582**±0.01	**3.1045**±0.21	**0.6582**±0.14	**0.9999**±0.00	**0.0020**±0.00	**0.1165**±0.02

Algorithms	Horizontal Translation				Pure Rotation			
	R_{sp} ↑	ACS ↑	AEPE ↓	AAE (°) ↓	R_{sp} ↑	ACS ↑	AEPE ↓	AAE (°) ↓
Horn-Schunck [20]	0.9154±0.11	0.9962±0.00	0.0241±0.01	1.2504±0.25	0.7021±0.09	0.7305±0.01	1.0852±0.01	42.0120±4.15
Lucas-Kanade [31]	0.6205±0.18	0.9985±0.00	0.0165±0.00	0.8621±0.18	0.4589±0.14	0.9964±0.00	0.1235±0.01	3.7145±0.08
Farneback [14]	**0.9485**±0.06	**0.9999**±0.00	0.0075±0.00	0.3910±0.08	0.6924±0.10	0.9971±0.00	0.1082±0.01	3.4250±0.06
Brox [6]	0.8950±0.15	0.9995±0.00	0.0112±0.00	0.5621±0.12	0.8582±0.08	0.9975±0.00	0.1271±0.01	3.4102±0.07
TV-L1 [64]	0.8675±0.12	0.9992±0.00	0.0102±0.00	0.5422±0.14	0.5235±0.13	0.9968±0.00	0.1050±0.01	3.3055±0.11
DeepFlow [58]	0.3905±0.21	0.9989±0.00	0.0275±0.00	1.4510±0.31	0.6558±0.13	0.9981±0.00	0.0845±0.01	2.6512±0.08
PCAFlow [61]	0.5125±0.23	0.9972±0.00	0.0381±0.01	2.0215±0.48	0.1685±0.08	0.9925±0.01	0.1945±0.01	5.8420±0.12
DIS [27]	0.8621±0.16	0.9991±0.00	0.0115±0.00	0.5512±0.11	0.4610±0.09	0.9969±0.00	0.1095±0.01	3.4510±0.07
RPKNet [35]	0.3485±0.25	0.9974±0.00	0.0441±0.01	2.3510±0.52	0.0791±0.11	0.9912±0.01	0.2485±0.01	7.0325±0.11
SEA-RAFT (M) [57]	0.2745±0.28	0.9971±0.00	0.0565±0.01	3.0750±0.68	0.0415±0.09	0.9915±0.01	0.1925±0.02	6.1050±0.52
DPFlow [34]	0.4855±0.24	0.9976±0.00	0.0472±0.01	2.5910±0.65	0.1145±0.12	0.9942±0.01	0.1785±0.01	5.5510±0.10
ReynoldsFlow (Ours)	0.9402±0.08	0.9998±0.00	**0.0064**±0.00	**0.3651**±0.05	**0.9015**±0.03	**0.9993**±0.00	**0.0615**±0.01	**1.9840**±0.07

4.3 Action Recognition on HMDB51 and UCF101

Beyond pose estimation, we evaluate action recognition on two widely used benchmarks: HMDB51 [28], containing 6,766 clips across 51 action categories (resolutions 176×240–592×240), and UCF101 [48], with 13,320 clips across 101 categories at 320×240. Following the self-supervised video representation framework of [23], we adopt their C3D-based architecture with different flow inputs, keeping the original training configuration (batch size 6, 100 epochs pre-training, 75 epochs fine-tuning). Classification accuracy is reported using two-fold cross-validation. As shown in Tab. 3, ReynoldsFlow consistently improves over the RGB baseline and performs comparably to DL-based methods, even if it does not achieve the highest absolute accuracy.

4.4 Object Detection on UAV Datasets

We evaluate ReynoldsFlow on UAV object detection using three datasets: Anti-UAV, ARD100, and UAVDB. Anti-UAV contains infrared frames (512×512 to 640×512) captured by a moving camera; ARD100 comprises high-resolution RGB videos (1920×1080) from Phantom-type drones; UAVDB includes high-resolution RGB frames (1920×1080 to 3840×2160) from a static ground cam-

Table 3: Effect of velocity fields estimation on downstream task performance across GolfDB [33], HMDB51 [28], UCF101 [48], Anti-UAV [24], ARD100 [18], and UAVDB [8]. Best results are in **bold**, second best are underlined.

Methods	GolfDB	HMDB51	UCF101	Anti-UAV		ARD100		UAVDB	
	PCE ↑	Accuracy ↑	Accuracy ↑	AP_{50}^{test} ↑	AP_{50-95}^{test} ↑	AP_{50}^{test} ↑	AP_{50-95}^{test} ↑	AP_{50}^{test} ↑	AP_{50-95}^{test} ↑
RGB	0.705	0.372	0.698	–		<u>0.554</u>	<u>0.304</u>	<u>0.811</u>	<u>0.518</u>
Grayscale / Infrared	0.698	0.328	0.614	<u>0.781</u>	<u>0.418</u>	0.376	0.167	0.660	0.281
Horn-Schunck [20]	0.707	0.165	0.259	0.217	0.080	0.074	0.016	0.104	0.021
Lucas-Kanade [31]	0.692	0.214	0.338	0.280	0.114	0.237	0.141	0.500	0.200
Farneback [14]	0.717	0.248	0.383	0.500	0.246	0.182	0.103	0.258	0.145
Brox [6]	0.708	0.260	0.328	0.379	0.168	0.122	0.089	0.244	0.110
TV-L1 [64]	<u>0.810</u>	0.284	0.537	0.600	0.278	0.227	0.127	0.779	0.409
DeepFlow [58]	0.706	0.237	0.419	0.344	0.143	0.138	0.063	0.154	0.058
PCAFlow [61]	0.765	0.296	0.424	0.580	0.286	0.128	0.041	0.547	0.332
DIS [27]	0.772	0.207	0.562	0.347	0.144	0.102	0.018	0.151	0.057
RPKNet [35]	0.801	0.395	**0.734**	0.316	0.214	0.088	0.014	0.110	0.039
SEA-RAFT (M) [57]	0.782	**0.419**	<u>0.722</u>	0.357	0.188	0.089	0.048	0.486	0.243
DPFlow [34]	0.786	<u>0.411</u>	0.705	0.427	0.265	0.072	0.016	0.270	0.101
ReynoldsFlow (Ours)	**0.812**	0.402	0.714	**0.792**	**0.446**	**0.602**	**0.326**	**0.895**	**0.547**

era. All datasets focus on single-class UAVs at varying scales. We sample 4,800/1,600/1,600 images for training/validation/test from Anti-UAV, approximately 12,000/4,000/4,000 from ARD100, and use the official split for UAVDB.

Flow estimates from all methods are fed into YOLOv11n [26], trained with a batch size of 64, 640×640 input resolution, eight workers, and over 100 epochs. Mosaic augmentation is applied except for the final ten epochs, and pre-trained weights are fine-tuned for each dataset. Performance is reported using AP_{50} and AP_{50-95}. As shown in Tab. 3, ReynoldsFlow consistently improves YOLOv11n across all datasets. Beyond Tab. 3, ReynoldsFlow achieves comparable or higher accuracy than spatiotemporal embedding models, such as TransVisDrone [45], which are specifically designed for UAV detection. However, these models require over five times longer training and more than six times slower inference. In contrast, ReynoldsFlow adds only negligible overhead relative to model inference, avoids task-specific retraining, and seamlessly integrates as a plug-and-play representation for diverse downstream tasks.

4.5 Analysis and Discussion

ReynoldsFlow demonstrates clear advantages in accuracy, efficiency, and representation design. Large objects benefit from directional cues, yielding predictable gains across flow features. The strength of ReynoldsFlow is especially pronounced for tiny or nearly invisible targets, where directional information is less informative but motion magnitude remains critical, allowing ReynoldsFlow to maintain high accuracy where both classical and DL methods struggle.

Our ablation study provides three key insights. First, the proposed ReynoldsFlow representation consistently outperforms conventional HSV-based visual-

Table 4: Ablation study of ReynoldsFlow representations on downstream task performance. Best results are in **bold**, second best are underlined.

Representations	GolfDB	HMDB51	UCF101	Anti-UAV		ARD100		UAVDB	
	PCE ↑	Accuracy ↑	Accuracy ↑	AP_{50}^{test} ↑	AP_{50-95}^{test} ↑	AP_{50}^{test} ↑	AP_{50-95}^{test} ↑	AP_{50}^{test} ↑	AP_{50-95}^{test} ↑
HSV [30]	0.804	0.382	0.597	0.646	0.320	0.417	0.211	0.500	0.288
$[\lvert\boldsymbol{v}_d^n\rvert, -, f^n]$	0.788	0.375	0.684	0.765	0.407	0.478	0.262	0.803	0.471
$[\lvert\boldsymbol{v}_c^n\rvert, \lvert\boldsymbol{v}_d^n\rvert, f^n]$	0.791	0.367	0.699	0.784	0.386	0.509	0.287	0.831	0.492
$[\lvert\boldsymbol{v}_d^n\rvert, \lvert\boldsymbol{v}_c^n\rvert, f^n]$	**0.812**	**0.402**	**0.714**	**0.792**	**0.446**	**0.602**	**0.326**	**0.895**	**0.547**

izations by omitting directional encoding, as shown in the first and last rows of Tab. 4. Second, incorporating the CF magnitude $|\boldsymbol{v}_c^n|$ consistently improves performance, as demonstrated by the second and last rows. Third, among six channel-ordering configurations, assigning motion magnitude to the green channel achieves the best performance, consistent with the RGGB Bayer pattern, in which the green channel has the highest sampling density and contributes most to luminance perception. These findings highlight the importance of the CF component and the proposed feature representation.

5 Conclusion

In this paper, we presented ReynoldsFlow, a physics-inspired, zero-shot video representation that overcomes the brightness constancy limitation in motion estimation. By grounding our formulation in the RTT and HHD, we decompose video motion into CF and DF components, yielding a divergence-compensated solver that eliminates tracking artifacts caused by geometric zooming and varying illumination. Quantitative results show that ReynoldsFlow consistently outperforms classical and DL-based OF methods in challenging scenarios while running in real time. As a robust, plug-and-play representation, it stacks the CF and DF magnitudes to enhance downstream tasks, including pose estimation, action recognition, and tiny UAV detection, without task-specific retraining or costly 3D convolutions. ReynoldsFlow demonstrates that transparent, physics-grounded modeling can provide an interpretable, efficient, and powerful alternative to purely data-driven architectures for video understanding.

Acknowledgements

This work was supported in part by the resources provided by National Yang Ming Chiao Tung University and The University of Melbourne. The authors also acknowledge the developers of the open-source software and the providers of the publicly available datasets that made this research possible. The corresponding author gratefully acknowledges the support from the National Science and Technology Council (NSTC), Taiwan, under Grant No. 114-2115-M-008-007-MY2.

References

1. Agbinya, J.I., Rees, D.: Multi-object tracking in video. Real-Time Imaging **5**(5), 295–304 (1999)
2. Arfken, G.B., Weber, H.J., Harris, F.E.: Mathematical methods for physicists: a comprehensive guide. Academic press (2011)
3. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International journal of computer vision **92**, 1–31 (2011)
4. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. International journal of computer vision **12**(1), 43–77 (1994)
5. Bhatia, H., Norgard, G., Pascucci, V., Bremer, P.T.: The helmholtz-hodge decomposition—a survey. IEEE Transactions on visualization and computer graphics **19**(8), 1386–1404 (2012)
6. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV 8. pp. 25–36. Springer (2004)
7. Charles, J., Pfister, T., Magee, D., Hogg, D., Zisserman, A.: Personalizing human video pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3063–3072 (2016)
8. Chen, Y.H.: Uavdb: Trajectory-guided adaptable bounding boxes for uav detection. arXiv preprint arXiv:2409.06490 (2024)
9. Chen, Y.H.: Strong baseline: Multi-uav tracking via yolov12 with bot-sort-reid. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6573–6582 (2025)
10. Ciaparrone, G., Sánchez, F.L., Tabik, S., Troiano, L., Tagliaferri, R., Herrera, F.: Deep learning in video multi-object tracking: A survey. Neurocomputing **381**, 61–88 (2020)
11. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
12. Ebrahimi Kahou, S., Michalski, V., Konda, K., Memisevic, R., Pal, C.: Recurrent neural networks for emotion recognition in video. In: Proceedings of the 2015 ACM on international conference on multimodal interaction. pp. 467–474 (2015)
13. Fadili, M.J., Starck, J.L., Bobin, J., Moudden, Y.: Image decomposition and separation using sparse representations: An overview. Proceedings of the IEEE **98**(6), 983–994 (2009)
14. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13. pp. 363–370. Springer (2003)
15. Gai, D., Feng, R., Min, W., Yang, X., Su, P., Wang, Q., Han, Q.: Spatiotemporal learning transformer for video-based human pose estimation. IEEE Transactions on Circuits and Systems for Video Technology **33**(9), 4564–4576 (2023)
16. Girdhar, R., Gkioxari, G., Torresani, L., Paluri, M., Tran, D.: Detect-and-track: Efficient pose estimation in videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 350–359 (2018)
17. Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks pp. 37–45 (2012)

18. Guo, H., Lin, X., Zhao, S.: Yolomg: Vision-based drone-to-drone detection with appearance and pixel-level motion fusion. arXiv preprint arXiv:2503.07115 (2025)
19. Guo, J., Zhang, A., Remias, E., Sheikholeslami, G.: Image decomposition and representation in large image database systems. Journal of Visual Communication and Image Representation **8**(2), 167–181 (1997)
20. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981)
21. Hsu, T.C., Liao, Y.S., Huang, C.R.: Video summarization with spatiotemporal vision transformer. IEEE Transactions on Image Processing **32**, 3013–3026 (2023)
22. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017)
23. Jenni, S., Meishvili, G., Favaro, P.: Video representation learning by recognizing temporal transformations. In: European conference on computer vision. pp. 425–442. Springer (2020)
24. Jiang, N., Wang, K., Peng, X., Yu, X., Wang, Q., Xing, J., Li, G., Ye, Q., Jiao, J., Han, Z., et al.: Anti-uav: a large-scale benchmark for vision-based uav tracking. T-MM (2021)
25. Jiao, L., Zhang, R., Liu, F., Yang, S., Hou, B., Li, L., Tang, X.: New generation deep learning for video object detection: A survey. IEEE Transactions on Neural Networks and Learning Systems **33**(8), 3195–3215 (2021)
26. Jocher, G., Qiu, J.: Ultralytics yolo11 (2024), `https://github.com/ultralytics/ultralytics`
27. Kroeger, T., Timofte, R., Dai, D., Van Gool, L.: Fast optical flow using dense inverse search. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 471–488. Springer (2016)
28. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: 2011 International conference on computer vision. pp. 2556–2563. IEEE (2011)
29. Li, Z., Zhao, M., Yang, X., Liu, Y., Sheng, J., Zeng, X., Wang, T., Wu, K., Jiang, Y.G.: Stnmamba: Mamba-based spatial-temporal normality learning for video anomaly detection. arXiv preprint arXiv:2412.20084 (2024)
30. Liu, C., et al.: Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. thesis, Massachusetts Institute of Technology (2009)
31. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th international joint conference on Artificial intelligence. vol. 2, pp. 674–679 (1981)
32. Madake, J., Sirshikar, S., Kulkarni, S., Bhatlawande, S.: Golf shot swing recognition using dense optical flow. In: 2023 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS). pp. 1–5. IEEE (2023)
33. McNally, W., Vats, K., Pinto, T., Dulhanty, C., McPhee, J., Wong, A.: Golfdb: A video database for golf swing sequencing. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 2553–2562 (2019). `https://doi.org/10.1109/CVPRW.2019.00311`
34. Morimitsu, H., Zhu, X., Cesar Jr, R.M., Ji, X., Yin, X.C.: Dpflow: Adaptive optical flow estimation with a dual-pyramid framework. arXiv preprint arXiv:2503.14880 (2025)

35. Morimitsu, H., Zhu, X., Ji, X., Yin, X.C.: Recurrent partial kernel network for efficient optical flow estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 4278–4286 (2024)
36. Palit, B., Basu, A., Mandal, M.K.: Applications of the discrete hodge helmholtz decomposition to image and video processing. In: International Conference on Pattern Recognition and Machine Intelligence. pp. 497–502. Springer (2005)
37. Park, J., Kim, H.S., Ko, K., Kim, M., Kim, C.: Videomamba: Spatio-temporal selective state space model. In: European Conference on Computer Vision. pp. 1–18. Springer (2024)
38. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7753–7762 (2019)
39. Peng, B., Lei, J., Fu, H., Jia, Y., Zhang, Z., Li, Y.: Deep video action clustering via spatio-temporal feature learning. Neurocomputing **456**, 519–527 (2021)
40. Qi, M., Idoughi, R., Heidrich, W.: Hdnet: physics-inspired neural network for flow estimation based on helmholtz decomposition. arXiv preprint arXiv:2406.08570 (2024)
41. Qiu, Z., Yao, T., Mei, T.: Learning deep spatio-temporal dependence for semantic video segmentation. IEEE Transactions on Multimedia **20**(4), 939–949 (2017)
42. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4161–4170 (2017)
43. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1164–1172 (2015)
44. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
45. Sangam, T., Dave, I.R., Sultani, W., Shah, M.: Transvisdrone: Spatio-temporal transformer for vision-based drone-to-drone detection in aerial videos. arXiv preprint arXiv:2210.08423 (2022)
46. Schwarz, G.: Hodge Decomposition-A method for solving boundary value problems. Springer (2006)
47. Shih, C.C., Tyan, H.R., Liao, H.M.: Shot change detection based on the reynolds transport theorem. In: Pacific-Rim Conference on Multimedia. pp. 819–824. Springer (2001)
48. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
49. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 2432–2439. IEEE (2010)
50. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
51. Sun, Y., Zhi, X., Han, H., Jiang, S., Shi, T., Gong, J., Zhang, W.: Enhancing uav detection in surveillance camera videos through spatiotemporal information and optical flow. Sensors **23**(13), 6037 (2023)
52. Tao, M., Bai, J., Kohli, P., Paris, S.: Simpleflow: A non-iterative, sublinear optical flow algorithm. In: Computer graphics forum. vol. 31, pp. 345–353. Wiley Online Library (2012)

53. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020)
54. Tran, T.M., Bui, D.C., Nguyen, T.V., Nguyen, K.: Transformer-based spatio-temporal unsupervised traffic anomaly detection in aerial videos. IEEE Transactions on Circuits and Systems for Video Technology **34**(9), 8292–8309 (2024)
55. Ul Amin, S., Kim, B., Jung, Y., Seo, S., Park, S.: Video anomaly detection utilizing efficient spatiotemporal feature fusion with 3d convolutions and long short-term memory modules. Advanced Intelligent Systems **6**(7), 2300706 (2024)
56. Von Marcard, T., Pons-Moll, G., Rosenhahn, B.: Human pose estimation from video and imus. IEEE transactions on pattern analysis and machine intelligence **38**(8), 1533–1547 (2016)
57. Wang, Y., Lipson, L., Deng, J.: Sea-raft: Simple, efficient, accurate raft for optical flow. In: European Conference on Computer Vision. pp. 36–54. Springer (2024)
58. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE international conference on computer vision. pp. 1385–1392 (2013)
59. White, F.: Fluid Mechanics. McGraw-Hill series in mechanical engineering, McGraw Hill (2011), `https://books.google.com.au/books?id=egk8SQAACAAJ`
60. Wu, J., Zhang, B., Xu, Y., Zhang, D.: Illuminance compensation and texture enhancement via the hodge decomposition. IEEE Transactions on Circuits and Systems for Video Technology **31**(3), 956–971 (2020)
61. Wulff, J., Black, M.J.: Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 120–130 (2015)
62. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. arXiv preprint arXiv:1712.04851 **1**(2), 5 (2017)
63. Yuan, J., Schnörr, C., Steidl, G.: Convex hodge decomposition and regularization of image flows. Journal of Mathematical Imaging and Vision **33**(2), 169–177 (2009)
64. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007. Proceedings 29. pp. 214–223. Springer (2007)
65. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: European conference on computer vision. pp. 766–782. Springer (2016)
66. Zhao, B., Li, X., Lu, X.: Hierarchical recurrent neural network for video summarization. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 863–871 (2017)
67. Zhu, H., Wei, H., Li, B., Yuan, X., Kehtarnavaz, N.: A review of video object detection: Datasets, metrics and methods. Applied Sciences **10**(21), 7834 (2020)
68. Zhu, X., Dai, J., Yuan, L., Wei, Y.: Towards high performance video object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7210–7218 (2018)
69. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 408–417 (2017)
70. Zhu, Z., Yin, H., Chai, Y., Li, Y., Qi, G.: A novel multi-modality image fusion method based on image decomposition and sparse representation. Information Sciences **432**, 516–529 (2018)