REYNOLDSFLOW: SPATIOTEMPORAL FLOW REPRESENTATIONS FOR VIDEO LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation learning for videos has largely relied on spatiotemporal modules embedded in deep architectures, which, while effective, often require heavy computation and heuristic design. Existing approaches, such as 3D convolutional modules or optical flow networks, may also overlook changes in illumination, scale variations, and structural deformations in video sequences. To address these challenges, we propose ReynoldsFlow, a physics-inspired flow representation that leverages the Helmholtz decomposition and the Reynolds transport theorem to derive principled spatiotemporal features directly from video data. Unlike classical optical flow, ReynoldsFlow captures both divergence-free and curl-free components under more general assumptions, enabling robustness to photometric variation while preserving intrinsic structure. Beyond its theoretical grounding, ReynoldsFlow remains lightweight and adaptable, combining frame intensity with flow magnitude to yield texture-preserving and dynamics-aware representations that substantially enhance tiny object detection. Experiments on benchmarks with various target scales demonstrate that ReynoldsFlow is consistently comparable to or outperforms existing flow-based features, while also improving interpretability and efficiency. These results position ReynoldsFlow as a compelling representation for video understanding and a strong foundation for downstream model learning. The code will be made publicly available.

1 INTRODUCTION

Computer vision techniques are now deeply embedded in everyday life, with video functionality serving as a core feature of modern smartphones. Beyond casual use, video understanding underpins a broad spectrum of applications, including stabilization, interpolation, object detection (Jiao et al., 2021; Zhu et al., 2018a; 2020; 2017), multi-object tracking (Ciaparrone et al., 2020; Agbinya & Rees, 1999; Chen, 2025), and pose estimation (Girdhar et al., 2018; Charles et al., 2016; Von Marcard et al., 2016; Pavllo et al., 2019). Progress in these areas has been largely driven by deep learning architectures such as recurrent neural networks (RNNs) (Zhao et al., 2017; Ebrahimi Kahou et al., 2015), Long Short-Term Memory (LSTM) networks (Graves, 2012; Zhang et al., 2016), and, more recently, transformer- and Mamba-based spatiotemporal models (Tran et al., 2024; Hsu et al., 2023; Gai et al., 2023; Zhang et al., 2021; Li et al., 2024; Park et al., 2024). Despite their strong empirical performance, these methods face notable limitations: they often incur high computational costs, rely on carefully tuned architectural heuristics, and offer limited interpretability, as their representations are rarely grounded in the physical principles of motion.

To address these challenges, we introduce ReynoldsFlow, a physics-inspired representation for video learning. Building on optical flow estimation (Sun et al., 2010), Helmholtz decomposition (Arfken et al., 2011), and the Reynolds transport theorem (RTT) (White, 2011), ReynoldsFlow provides an interpretable analogue to modern spatiotemporal modules, bridging the gap between data-driven architectures and physically principled modeling. By directly deriving video representations from data, ReynoldsFlow reduces dependence on heuristic design, improves computational efficiency, and produces more robust and generalizable spatiotemporal features. Importantly, it operates in a training-free, unsupervised manner, avoiding the need for large labeled datasets or costly optimization. Our main contributions are summarized as follows:

1. **ReynoldsFlow representation**. We introduce ReynoldsFlow, an unsupervised spatiotemporal flow representation derived from the Helmholtz decomposition and the RTT. ReynoldsFlow jointly captures divergence-free and curl-free components, providing a physically interpretable motion descriptor under general assumptions.

2. **Efficiency and plug-and-play adaptability**. We demonstrate that ReynoldsFlow achieves accuracy comparable to SOTA spatiotemporal embedding models while being significantly more computationally efficient. Furthermore, it can be directly computed from raw video data without retraining or task-specific preprocessing, making it a lightweight plug-and-play representation for downstream neural networks.

3. **Comprehensive validation**. We conduct extensive experiments across multiple video datasets, showing that ReynoldsFlow is both highly interpretable and effective in enhancing diverse tasks, including pose estimation, action recognition, and object detection.

2 Related Work

2.1 Optical Flow and Video Motion Estimation

Early optical flow (OF) methods such as Horn-Schunck (Horn & Schunck, 1981) and Lucas–Kanade (Lucas & Kanade, 1981) laid the foundation for motion estimation. Horn–Schunck enforced global smoothness constraints but struggled with large displacements and motion boundaries. In contrast, Lucas–Kanade offered computational efficiency but was limited to small, consistent motions and performed poorly in complex or textured scenes. Subsequent refinements, including Optimal Filter Estimation (Sharmin & Brad, 2012) and hybrid methods combining Horn–Schunck and Lucas–Kanade (Bruhn et al., 2005), aimed to balance adaptability and accuracy but remained sensitive to occlusions, noise, and background texture. Advancements such as Farneback's polynomial expansion (Farnebäck, 2003) and Brox's high-accuracy variational method (Brox et al., 2004) improved motion estimation at the cost of increased computational demand. TV-L1 (Zach et al., 2007) introduced noise resilience through total variation regularization, while SimpleFlow (Tao et al., 2012) provided a non-iterative approach, trading off accuracy for speed. RLOF (Senst et al., 2012) advanced feature tracking robustness but remained challenged by high-texture regions. To improve efficiency without sacrificing accuracy, methods like DeepFlow (Weinzaepfel et al., 2013) used deep matching, PCAFlow (Wulff & Black, 2015) applied low-dimensional motion bases, and DIS (Kroeger et al., 2016) optimized dense flow for time-critical tasks.

Deep learning reshaped OF, beginning with FlowNet (Dosovitskiy et al., 2015), the first CNN-based model, which improved recognition but struggled with fine-grained motion. FlowNet2 (Ilg et al., 2017) addressed this with a cascaded refinement architecture. EpicFlow (Revaud et al., 2015) combined sparse matching with variational refinement to improve boundary localization. PWC-Net (Sun et al., 2018) introduced a pyramid, warping, and cost-volume framework for practical multiscale motion estimation, while SpyNet (Ranjan & Black, 2017) focused on computational efficiency for real-time use. RAFT (Teed & Deng, 2020) achieved SOTA performance by iteratively refining dense flow fields using learned contextual correlations. Recent extensions such as RPKNet (Morimitsu et al., 2024), SEA-RAFT (Wang et al., 2024), and DPFlow (Morimitsu et al., 2025) further enhanced temporal consistency and scene understanding through recurrent and domain-adaptive mechanisms. Despite improved accuracy, deep learning-based methods remain data-hungry, computationally expensive, and less practical for out-of-the-box deployment.

A task similar to our UAV detection scenario is explored in (Sun et al., 2023), where a convolutional layer emulating the Lucas-Kanade OF is integrated into the YOLO architecture to enhance UAV detection. Similarly, in (Madake et al., 2023), the authors employ Farneback OF to track a golfer's body posture and swing trajectory, followed by handcrafted feature-based classification. Despite their ingenuity, both approaches suffer from the limited informativeness of conventional OF visualizations, making them less effective for fine-grained motion understanding. To our knowledge, no existing work has shown substantial improvements in object detection or pose estimation by directly incorporating OF images into deep neural networks.

2.2 Decomposition-based Representations

The Helmholtz–Hodge decomposition (HHD) provides a principled mathematical framework for separating a vector field into divergence-free, curl-free, and harmonic components (Schwarz, 2006; Bhatia et al., 2012). 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 (Palit et al., 2005). To enhance robustness, convex optimization formulations were later introduced to regularize image flows and stabilize the decomposition process (Yuan et al., 2008; 2009). Beyond flow fields, decomposition has also played an important role in image representation more broadly, ranging from structural organization in large-scale image databases (Guo et al., 1997), to sparse decomposition methods for separating signal components (Fadili et al., 2009), and multi-modality image fusion based on decomposition and sparse coding (Zhu et al., 2018b).

More recently, decomposition-based techniques have been extended to practical vision tasks, such as illumination compensation and texture enhancement in imaging (Wu et al., 2020). Physics-inspired neural architectures further push this line of research, as in HDNet (Qi et al., 2024), which leverages Helmholtz decomposition to design interpretable deep networks for flow estimation. Collectively, these works demonstrate 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 vector 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 essential for effective video understanding, as motion information provides critical cues for object behavior, scene changes, and activity recognition. Early deep learning approaches extended convolutional networks to the spatiotemporal domain using 3D convolutions and recurrent modules to capture temporal dependencies across frames (Xie et al., 2017; Qiu et al., 2017; Peng et al., 2021; Ul Amin et al., 2024). Such spatiotemporal feature learning has been applied to diverse tasks, including semantic video segmentation (Qiu et al., 2017), action clustering (Peng et al., 2021), video deraining (Zhang et al., 2022), and anomaly detection (Ul Amin et al., 2024), demonstrating the broad applicability of learned temporal representations. While effective, these approaches typically rely on supervised training with large annotated datasets or costly optimization, limiting their flexibility and interpretability.

Complementary to purely data-driven methods, physics-inspired strategies provide alternative insights into temporal modeling. Specifically, as presented in (Shih et al., 2001), the authors applied the RTT to treat a video clip as a continuous frame flow, enabling theoretically grounded detection of temporal changes such as shot transitions. In contrast, our approach builds on the same underlying principle but integrates it with flow components separated via the HHD to analyze temporal evolution more comprehensively. This combination allows ReynoldsFlow to directly extract interpretable, spatiotemporal video representations from data in an unsupervised, training-free manner, rather than being restricted to a single task. Consequently, our framework bridges the gap between principled physical modeling and modern spatiotemporal feature learning, producing robust representations that can support a wide range of downstream video understanding tasks.

3 ReynoldsFlow

In this section, we first introduce the necessary preliminaries and then formulate ReynoldsFlow using the Reynolds transport theorem (RTT) (White, 2011), offering a novel physics-inspired perspective on flow estimation. We subsequently present a dedicated visualization scheme to illustrate ReynoldsFlow features. Finally, we evaluate the runtime performance of our method, highlighting its computational efficiency compared to existing flow estimation approaches.

3.1 Preliminaries

3.1.1 Helmholtz decomposition

Let $\boldsymbol{v} \in C^1(\Omega, \mathbb{R}^2)$ be a continuously differentiable vector field defined on a domain Ω. According to the Helmholtz decomposition theorem (Arfken et al., 2011), $\boldsymbol{v}$ can be uniquely decomposed into the sum of an irrotational (curl-free) component and a solenoidal (divergence-free) component:

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

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

3.1.2 Reynolds transport theorem

Consider a smooth scalar function $f = f(\boldsymbol{p}, t)$ defined on a time-dependent domain $\Omega(t)$. A grayscale video can be represented as $f(\boldsymbol{p}(t^n), t^n)$, where f is the intensity at pixel location $\boldsymbol{p}(t^n)$ in the field of view $\Omega(t^n)$ of the camera, and t^n denotes the time at nth frame. The RTT (White, 2011) 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 vector 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 divergence-free (i.e. $\boldsymbol{v} = \boldsymbol{v}_d$) for all time, then equation 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 equation 3 vanishes pointwise and $\boldsymbol{v}_d$ is piecewise constant, equation 3 reduces to the traditional optical flow (OF). We denote the velocity field satisfying equation 2 as $\boldsymbol{v}_r$, referred to as ReynoldsFlow. To clarify its relation with OF, we rename the divergence-free flow $\boldsymbol{v}_d$ as $\boldsymbol{v}_o$, while the curl-free component $\boldsymbol{v}_c$ serves as the complementary flow (CF).

3.1.3 Area Jacobian in domain transformation

Consider a region $\Omega(t)$ evolving over time t under the influence of a vector 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 vector field $\boldsymbol{v}^n = \begin{pmatrix} v_x^n \\ v_y^n \end{pmatrix}$, and assume an 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 equation 4, the differential wedge product $dx \wedge dy$ at $\boldsymbol{p}^{n+1}$ can be approximated 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 Derivation of ReynoldsFlow

Recall that $\boldsymbol{v}_o$ denote the traditional OF satisfying the brightness constancy assumption:

$$\frac{\partial f}{\partial t} + \nabla f \cdot \boldsymbol{v}_o = 0,$$

where $\boldsymbol{v}_o$ is assumed piecewise constant over local patches $\omega(t)$. This leads to the classical OF:

$$\boldsymbol{v}_o = -(\nabla f)^{\dagger} \frac{\partial f}{\partial t}, \tag{7}$$

forming the basis of methods such as Lucas-Kanade and Horn-Schunck. To generalize beyond the brightness constancy and divergence-free assumptions, we invoke the RTT on each local patch $\omega(t)$ combined with the Helmholtz decomposition equation 1 to rewrite:

$$\frac{d}{dt}\int_{\omega(t)} f\,dA = \int_{\omega(t)} \left(\frac{\partial f}{\partial t} + \nabla f \cdot \boldsymbol{v}_o + \nabla f \cdot \boldsymbol{v}_c + f\nabla\cdot\boldsymbol{v}_c\right) dA. \tag{8}$$

Applying the explicit Euler method, the left-hand side (LHS) of equation 8 can be approximated as:

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

Next, using the Taylor approximation:

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

and the Helmholtz decomposition of the vector field $\boldsymbol{v}$,

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

Similarly, the right-hand side (RHS) of equation 8 is

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

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:

$$\int_{\partial\omega^n} \delta f^n \boldsymbol{v}_c^n \cdot \boldsymbol{n}\,dS^n - \int_{\omega^n} \nabla\delta f^n \cdot \boldsymbol{v}_c^n\,dA^n = 0. \tag{9}$$

As usual, we can compute the vector field $\boldsymbol{v}_c^n$ from equation 9 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 equation 9, we apply Simpson's rule:

$$\int_{\partial\omega_{3\times3}^n} \delta f^n \boldsymbol{v}_c^n \cdot \boldsymbol{n}\,dS^n \approx \left[\delta f_{b,x}^n, \delta f_{b,y}^n\right] \cdot \boldsymbol{v}_c^n, \tag{10}$$

where

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

The domain integral term in equation 9 is approximated as:

$$\int_{\omega^n_{3x3}} \nabla\delta f^n \cdot \boldsymbol{v}^n_c \, dA^n = \int_{\omega^n} [(\nabla\delta f^n)_x, (\nabla\delta f^n)_y] \cdot \boldsymbol{v}^n_c \, dA^n \approx [(\nabla\delta f^n_\omega)_x, (\nabla\delta f^n_\omega)_y] \cdot \boldsymbol{v}^n_c, \quad (11)$$

where

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

From equation 9, equation 10 and equation 11, we obtain the irrotational flow field:

$$\boldsymbol{v}^n_c = [(\delta f^n_b)_x - (\nabla\delta f^n_\omega)_x, (\delta f^n_b)_y - (\nabla\delta f^n_\omega)_y]^\perp.$$

Recall that the irrotational vector field $\boldsymbol{v}^n_c$ complements the OF field $\boldsymbol{v}^n_o$ derived from the RTT. Notably, the OF component is also naturally canceled out in equation 9, allowing CF to isolate residual non-motion effects such as illumination changes and non-rigid deformation. Hence, we refer to $\boldsymbol{v}^n_c$ as the complementary component of the OF in the sense of Helmholtz decomposition. To ensure smoothness in practice, we define:

$$\boldsymbol{v}^n_c = \begin{bmatrix} G * (-(\delta f^n_b)_y + (\nabla\delta f^n_\omega)_y) \\ G * ((\delta f^n_b)_x - (\nabla\delta f^n_\omega)_x) \end{bmatrix}, \quad (12)$$

where G represents a Gaussian smoothing kernel. Finally, ReynoldsFlow is defined as $\boldsymbol{v}^n_{\mathrm{R}} = \boldsymbol{v}^n_o + \boldsymbol{v}^n_c$, where $\boldsymbol{v}^n_o$ is given by equation 7 and $\boldsymbol{v}^n_c$ by equation 12.

3.3 ReynoldsFlow Representation

Flow is commonly visualized using the HSV color space, where motion magnitude and direction are mapped to hue and saturation (Baker et al., 2011). However, the HSV-to-RGB transformation is highly nonlinear, often introducing perceptual inconsistencies, particularly in low-texture regions, complex illumination, or dynamic motion scenarios. Such limitations can hinder downstream tasks, including tiny object detection, and reduce the robustness of neural networks relying solely on HSV-based representations. To address this, we propose an alternative visualization that augments flow features with additional cues. Specifically, we stack flow magnitudes across three channels, defining the ReynoldsFlow representation as $\boldsymbol{F}^n_{\mathrm{R}} = [|\boldsymbol{v}^n_o|, |\boldsymbol{v}^n_c|, f^n]$, while omitting directional information. In this scheme, the red and green channels encode the magnitudes of the OF and CF, respectively, while the blue channel preserves the current frame's intensity f^n, enhancing spatial detail and contrast. This design improves flow clarity and robustness to visual ambiguity, particularly for tiny or fast-moving objects. As visualized in Figure 1, $\boldsymbol{F}^n_{\mathrm{R}}$ yields sharper features across diverse datasets.

4 Experimental Results

To evaluate the proposed ReynoldsFlow representation, we conducted experiments on three computer vision tasks: (1) pose estimation on GolfDB (McNally et al., 2019), (2) action recognition on HMDB51 (Kuehne et al., 2011) and UCF101 (Soomro et al., 2012), and (3) object detection on Anti-UAV (Jiang et al., 2021), ARD100 (Guo et al., 2025), and UAVDB (Chen, 2024). We compared ReynoldsFlow with 13 approaches, including 1) original RGB and 2) grayscale videos, classical OF methods, 3) Horn-Schunck (Horn & Schunck, 1981), 4) dense Lucas-Kanade (Lucas & Kanade, 1981), 5) Farneback (Farnebäck, 2003), 6) Brox (Brox et al., 2004), 7) TV-L1 (Zach et al., 2007), 8) DeepFlow (Weinzaepfel et al., 2013), 9) PCAFlow (Wulff & Black, 2015), and 10) DIS (Kroeger et al., 2016), as well as deep learning-based methods such as 11) RPKNet (Morimitsu et al., 2024), 12) SEA-RAFT (Wang et al., 2024), and 13) DPFlow (Morimitsu et al., 2025). OF methods were selected for their training-free nature, aligning with the design philosophy of ReynoldsFlow. Learning-based methods, while capable of higher performance with domain-specific tuning, often require ground truth for training or fine-tuning, which is unavailable in most datasets. Therefore, we used the official pretrained models with the best-reported performance for each method. All flow representations are visualized in the HSV color space using the same visualization scheme, as described in (Liu et al., 2009), where motion magnitude and direction are encoded through hue and saturation, respectively. For the ReynoldsFlow representation, defined as $\boldsymbol{F}^n_{\mathrm{R}} = [|\boldsymbol{v}^n_o|, |\boldsymbol{v}^n_c|, f^n]$, we visualize only the magnitude components and omit directional information, as discussed earlier. All experiments reported in Section 4 were conducted on a high-performance computing system (Meade et al., 2017) equipped with an NVIDIA H100 GPU with 80 GB of memory.

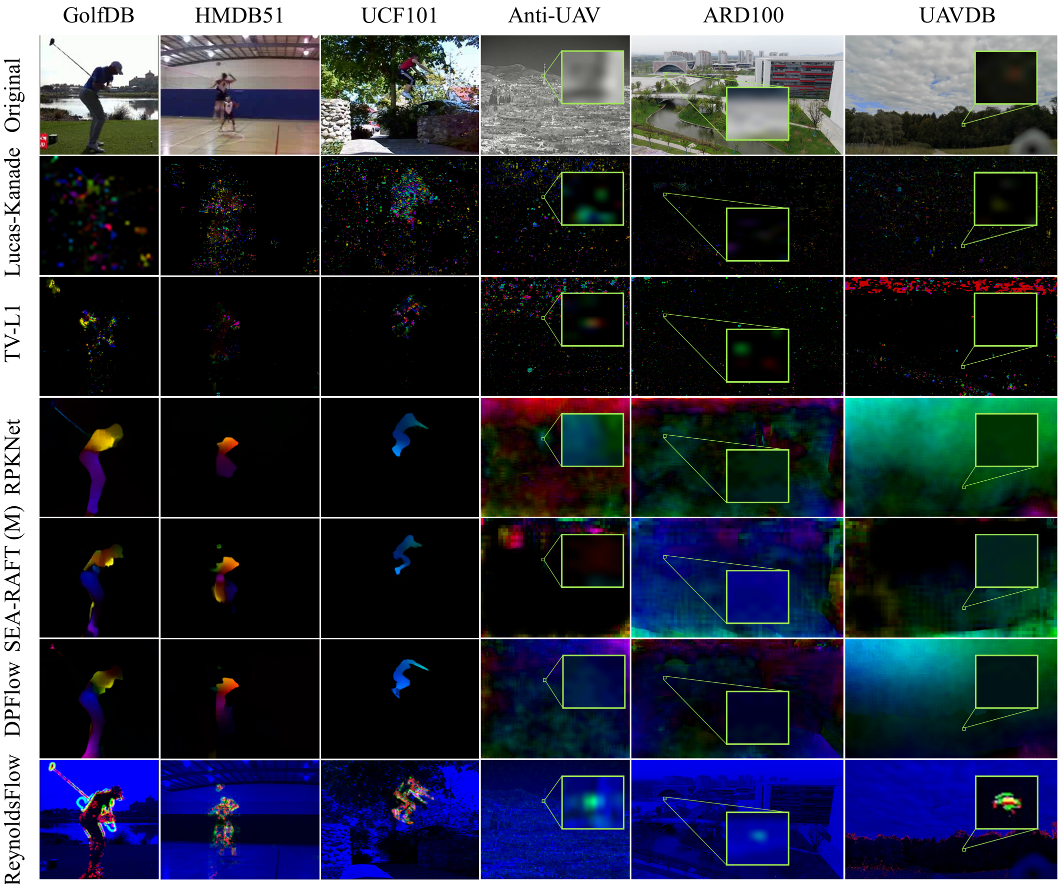

Figure 1: Top row (left to right): example frames from GolfDB (McNally et al., 2019), HMDB51 (Kuehne et al., 2011), UCF101 (Soomro et al., 2012), Anti-UAV (Jiang et al., 2021), ARD100 (Guo et al., 2025), and UAVDB (Chen, 2024). Middle rows: OF visualizations from Lucas-Kanade (Lucas & Kanade, 1981), TV-L1 (Zach et al., 2007), RPKNet (Morimitsu et al., 2024), SEA-RAFT (Wang et al., 2024), and DPFlow (Morimitsu et al., 2025), shown in HSV color encoding. Bottom row: ReynoldsFlow representations for each scene.

4.1 Pose Estimation on GolfDB

We evaluate the proposed method on the pose estimation task using the GolfDB dataset, which contains 1,400 videos at a resolution of 160×160 pixels. Each video features a subject who occupies nearly the entire frame. The objective is to accurately identify specific poses within the golf swing sequence, which is divided into eight distinct events: Address (A), Toe-up (TU), Mid-backswing (MB), Top (T), Mid-downswing (MD), Impact (I), Mid-follow-through (MFT), and Finish (F). The dataset includes face-on and down-the-line views, offering diverse perspectives to capture the nuanced transitions between each event.

For implementation, we adopt SwingNet (McNally et al., 2019), a model designed for pose estimation in golf swing videos. We maintain the original training configuration, including a sequence length of 64, 10 frozen layers, a batch size of 22, 2,000 iterations, and six workers. We use the pretrained MobileNetV2 model provided by (Sandler et al., 2018) and then fine-tune it on the GolfDB dataset. Performance is evaluated using the Percentage of Correct Events (PCE) metric, with four-fold cross-validation to ensure reliability. For real-time videos sampled at 30 fps, a tolerance of $\delta = 1$ is used. For slow-motion videos, the tolerance is calculated as $\delta = \max(\lfloor \frac{N}{f} \rceil, 1)$, where N is the number of frames from Address to Impact, f is the sampling frequency, and $\lfloor x \rceil$ denotes rounding x to the nearest integer. As shown in Table 1, our input achieves the highest PCE of 0.812.

Table 1: Effect of flow estimation on downstream task performance across GolfDB (McNally et al., 2019), HMDB51 (Kuehne et al., 2011), UCF101 (Soomro et al., 2012), Anti-UAV (Jiang et al., 2021), ARD100 (Guo et al., 2025), and UAVDB (Chen, 2024).

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	–		0.554	0.304	0.811	0.518
Grayscale / Infrared	0.698	0.328	0.614	0.781	0.418	0.376	0.167	0.660	0.281
Horn-Schunck (Horn & Schunck, 1981)	0.707	0.165	0.259	0.217	0.080	0.074	0.016	0.104	0.021
Lucas-Kanade (Lucas & Kanade, 1981)	0.692	0.214	0.338	0.280	0.114	0.237	0.141	0.500	0.200
Farneback (Farnebäck, 2003)	0.717	0.248	0.383	0.500	0.246	0.182	0.103	0.258	0.145
Brox (Brox et al., 2004)	0.708	0.260	0.328	0.379	0.168	0.122	0.089	0.244	0.110
TV-L1 (Zach et al., 2007)	0.810	0.284	0.537	0.600	0.278	0.227	0.127	0.779	0.409
DeepFlow (Weinzaepfel et al., 2013)	0.706	0.237	0.419	0.344	0.143	0.138	0.063	0.154	0.058
PCAFlow (Wulff & Black, 2015)	0.765	0.296	0.424	0.580	0.286	0.128	0.041	0.547	0.332
DIS (Kroeger et al., 2016)	0.772	0.207	0.562	0.347	0.144	0.102	0.018	0.151	0.057
RPKNet (Morimitsu et al., 2024)	0.801	0.395	**0.734**	0.316	0.214	0.088	0.014	0.110	0.039
SEA-RAFT (M) (Wang et al., 2024)	0.782	**0.419**	0.722	0.357	0.188	0.089	0.048	0.486	0.243
DPFlow (Morimitsu et al., 2025)	0.786	0.411	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**

4.2 Action Recognition on HMDB51 and UCF101

Beyond pose estimation, we also evaluate action recognition performance on two widely used benchmarks: HMDB51 (Kuehne et al., 2011), which contains 6,766 clips spanning 51 action categories with resolutions ranging from 176×240 to 592×240 pixels, and UCF101 (Soomro et al., 2012), which includes 13,320 clips at 320×240 resolution across 101 categories.

For implementation, we follow the self-supervised video representation learning framework of (Jenni et al., 2020), adopting their C3D-based architecture for action recognition with different flow inputs. We retain the original training configuration, including a batch size of 6, 100 epochs for pre-training, and 75 epochs for fine-tuning. Following standard practice, we report classification accuracy on both datasets using two-fold cross-validation for reliability. As shown in Table 1, although ReynoldsFlow does not achieve the highest accuracy on these benchmarks, it consistently improves over the RGB baseline and performs comparably to learning-based flow methods.

4.3 Object Detection on UAV Datasets

We further evaluate the proposed method on object detection tasks using three distinct UAV datasets: Anti-UAV, ARD100, and UAVDB. Anti-UAV consists of infrared video frames with resolution ranging from 512×512 to 640×512 captured by a moving camera. ARD100 comprises high-resolution RGB videos with resolution 1920×1080 featuring UAVs in diverse environments, with footage captured from Phantom-type drones. UAVDB is also composed of high-resolution RGB frames with resolution ranging from 1920×1080 to 3840×2160 recorded by a static ground camera. All three datasets focus on single-class UAV annotations and include targets at various scales, reflecting real-world, challenging scenarios where object size varies with distance from the camera. Particularly, we selected 4,800 training, 1,600 validation, and 1,600 test images from 223 Anti-UAV videos. For ARD100, which originally contains over 200,000 images, we sampled about one-tenth from each video, resulting in 12,000 training, 4,000 validation, and 4,000 test images from 100 videos. For the UAVDB dataset, we directly used the official split provided by the authors, which includes 10,763 training, 2,720 validation, and 4,578 test images.

In the implementation, we first compute flow estimations using the aforementioned methods, then apply YOLOv11n (Jocher & Qiu, 2024) as the object detector. All models were trained using eight workers, an input resolution of 640×640, a batch size of 64, and for over 100 epochs. Mosaic augmentation was applied throughout training except for the final ten epochs, during which it was disabled to stabilize convergence. We employed transfer learning by initializing from official YOLOv11n pre-trained weights and fine-tuning on the Anti-UAV, ARD100, and UAVDB datasets to incorporate prior knowledge. For evaluation, we report AP_{50} and AP_{50-95} on both validation and test sets. As demonstrated in Table 1, YOLOv11n with our input achieves the highest performance across all datasets, highlighting its effectiveness.

Beyond Table 1, we compare ReynoldsFlow with spatiotemporal embedding models such as TransVisDrone (Sangam et al., 2022), which are tailored for UAV detection. Although these models

Table 2: Runtime comparison (seconds) of OF algorithms on CPU or GPU for UAVDB per image.

Algorithms	OpenCV Packages	Runtime (s) ↓
Horn-Schunck (Horn & Schunck, 1981)	–	1.951
Lucas-Kanade (Lucas & Kanade, 1981)	cuda_DensePyrLKOpticalFlow	**0.013**
Farneback (Farnebäck, 2003)	cuda_FarnebackOpticalFlow	0.031
Brox (Brox et al., 2004)	cuda_BroxOpticalFlow	0.093
TV-L1 (Zach et al., 2007)	cuda_OpticalFlowDual_TVL1	3.165
DeepFlow (Weinzaepfel et al., 2013)	createOptFlow_DeepFlow	2.521
PCAFlow (Wulff & Black, 2015)	createOptFlow_PCAFlow	0.403
DIS (Kroeger et al., 2016)	DISOpticalFlow_create	0.046
RPKNet (Morimitsu et al., 2024)	–	1.568
SEA-RAFT (M) (Wang et al., 2024)	–	1.416
DPFlow (Morimitsu et al., 2025)	–	1.372
ReynoldsFlow (Ours)	–	0.019

Table 3: Ablation study of ReynoldsFlow representations on downstream task performance.

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 (Liu et al., 2009)	0.804	0.382	0.597	0.646	0.320	0.417	0.211	0.500	0.288
$[\lvert\boldsymbol{v}_o^n\rvert,\ -\ , f^n]$	0.788	0.375	0.684	0.765	0.407	0.478	0.262	0.803	0.471
$[\lvert\boldsymbol{v}_c^n\rvert, \lvert\boldsymbol{v}_o^n\rvert, f^n]$	0.791	0.367	0.699	0.784	0.386	0.509	0.287	0.831	0.492
$[\lvert\boldsymbol{v}_o^n\rvert, \lvert\boldsymbol{v}_c^n\rvert, f^n]$	**0.812**	**0.402**	**0.714**	**0.792**	**0.446**	**0.602**	**0.326**	**0.895**	**0.547**

achieve comparable accuracy, they require over five times longer training and more than six times slower inference. In contrast, ReynoldsFlow adds only negligible overhead relative to model inference, and a detailed runtime comparison is provided in Section 4.4. Moreover, spatiotemporal models demand task-specific preprocessing and retraining to adapt to new applications (e.g., pose estimation), whereas ReynoldsFlow is plug-and-play: the representation can be directly computed from raw video data and seamlessly integrated into downstream models.

4.4 Analysis and Discussion

ReynoldsFlow demonstrates clear benefits in accuracy, efficiency, and representation design. First, while large objects naturally benefit from accurate motion direction cues in video analysis and thus show predictable performance gains across different flow features, the advantage of ReynoldsFlow becomes particularly evident when dealing with tiny or nearly invisible targets. In these cases, directional information is less informative, but motion magnitude remains crucial, allowing ReynoldsFlow to maintain high accuracy where both classical and deep learning methods fail.

Second, ReynoldsFlow achieves strong runtime efficiency. On UAVDB, using an Intel Core i7-12650H CPU and an NVIDIA RTX 4050 GPU, it runs at 0.019 s per image, competitive with classical methods such as Farneback (0.031 s) and DIS (0.046 s), and substantially faster than more complex methods like Brox, TV-L1, and DeepFlow (0.09–3.2 s), and other learning-based approaches (around 1.5 s per image). A complete runtime comparison is provided in Table 2, demonstrating that ReynoldsFlow is well-suited for real-time and embedded applications.

Third, our ablation studies reveal three key insights: (1) **Visualization format:** The ReynoldsFlow representation (without directional information) substantially improves downstream video understanding performance over classical HSV color space visualization (with directional information), as shown by comparing the first and last rows of Table 3. (2) **CF magnitude contribution:** Incorporating the CF magnitude $\lvert\boldsymbol{v}_c^n\rvert$ consistently enhances the representation, evident from the comparison between the second and last rows in Table 3. (3) **Channel-ordering combinations:** Six combinations of stacked visualizations were evaluated, with the top two performances reported in the third and last rows in Table 3. Notably, placing the flow magnitude in the green channel outperforms placing the current frame's intensity there, which aligns with the RGGB Bayer filter pattern: the human eye is more sensitive to green light and perceives brightness more acutely than color detail.

These results highlight the benefits of incorporating the CF component into ReynoldsFlow and effectively combining representation channels. Overall, the experiments demonstrate that ReynoldsFlow provides robust video representations, runtime efficiency for resource-constrained deployment, and clear performance gains from incorporating CF.

5 Conclusion

We presented ReynoldsFlow, a physics-inspired video representation that captures both divergence-free and curl-free motion components in a principled, interpretable, and lightweight manner, grounded in the Helmholtz decomposition and Reynolds transport theorem. ReynoldsFlow delivers robust motion representations, improving over the RGB baseline on datasets with large objects and offering clear advantages for tiny or nearly invisible targets. It is plug-and-play, requiring no retraining or task-specific preprocessing. Ablation studies show that incorporating the CF magnitude and using stacked visualizations are crucial for improving performance across multiple datasets, highlighting both interpretability and practical utility. Importantly, potential applications of ReynoldsFlow include broader multi-task video learning scenarios and extensions to handle complex environmental variations, such as dynamic illumination and structural deformations. Moreover, due to its lightweight and efficient design, ReynoldsFlow is particularly well-suited for real-time deployment on resource-constrained platforms.

References

Johnson I Agbinya and David Rees. Multi-object tracking in video. *Real-Time Imaging*, 5(5): 295–304, 1999.

George B Arfken, Hans J Weber, and Frank E Harris. *Mathematical methods for physicists: a comprehensive guide*. Academic press, 2011.

Simon Baker, Daniel Scharstein, James P Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92:1–31, 2011.

Harsh Bhatia, Gregory Norgard, Valerio Pascucci, and Peer-Timo Bremer. The helmholtz-hodge decomposition—a survey. *IEEE Transactions on visualization and computer graphics*, 19(8): 1386–1404, 2012.

Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. 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.

Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61:211–231, 2005.

James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3063–3072, 2016.

Yu-Hsi Chen. Uavdb: Trajectory-guided adaptable bounding boxes for uav detection. *arXiv preprint arXiv:2409.06490*, 2024.

Yu-Hsi Chen. 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.

Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020.

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.

Samira Ebrahimi Kahou, Vincent Michalski, Kishore Konda, Roland Memisevic, and Christopher Pal. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pp. 467–474, 2015.

M Jalal Fadili, Jean-Luc Starck, Jérôme Bobin, and Yassir Moudden. Image decomposition and separation using sparse representations: An overview. *Proceedings of the IEEE*, 98(6):983–994, 2009.

Gunnar Farnebäck. 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.

Di Gai, Runyang Feng, Weidong Min, Xiaosong Yang, Pengxiang Su, Qi Wang, and Qing Han. Spatiotemporal learning transformer for video-based human pose estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):4564–4576, 2023.

Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 350–359, 2018.

Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Hanqing Guo, Xiuxiu Lin, and Shiyu Zhao. Yolomg: Vision-based drone-to-drone detection with appearance and pixel-level motion fusion. *arXiv preprint arXiv:2503.07115*, 2025.

J Guo, Aidong Zhang, Edward Remias, and Gholamhosein Sheikholeslami. Image decomposition and representation in large image database systems. *Journal of Visual Communication and Image Representation*, 8(2):167–181, 1997.

Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3): 185–203, 1981.

Tzu-Chun Hsu, Yi-Sheng Liao, and Chun-Rong Huang. Video summarization with spatiotemporal vision transformer. *IEEE Transactions on Image Processing*, 32:3013–3026, 2023.

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 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.

Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *European conference on computer vision*, pp. 425–442. Springer, 2020.

Nan Jiang, Kuiran Wang, Xiaoke Peng, Xuehui Yu, Qiang Wang, Junliang Xing, Guorong Li, Qixiang Ye, Jianbin Jiao, Zhenjun Han, et al. Anti-uav: a large-scale benchmark for vision-based uav tracking. *T-MM*, 2021.

Licheng Jiao, Ruohan Zhang, Fang Liu, Shuyuan Yang, Biao Hou, Lingling Li, and Xu Tang. New generation deep learning for video object detection: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3195–3215, 2021.

Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL `https://github.com/ultralytics/ultralytics`.

Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 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.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pp. 2556–2563. IEEE, 2011.

Zhangxun Li, Mengyang Zhao, Xuan Yang, Yang Liu, Jiamu Sheng, Xinhua Zeng, Tian Wang, Kewei Wu, and Yu-Gang Jiang. Stnmamba: Mamba-based spatial-temporal normality learning for video anomaly detection. *arXiv preprint arXiv:2412.20084*, 2024.

Ce Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pp. 674–679, 1981.

Jyoti Madake, Shreyas Sirshikar, Swagat Kulkarni, and Shripad Bhatlawande. 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.

William McNally, Kanav Vats, Tyler Pinto, Chris Dulhanty, John McPhee, and Alexander Wong. Golfdb: A video database for golf swing sequencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

Bernard Meade, Lev Lafayette, Greg Sauter, and Daniel Tosello. Spartan hpc-cloud hybrid: delivering performance and flexibility. *University of Melbourne*, 10:49, 2017.

Henrique Morimitsu, Xiaobin Zhu, Xiangyang Ji, and Xu-Cheng Yin. Recurrent partial kernel network for efficient optical flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4278–4286, 2024.

Henrique Morimitsu, Xiaobin Zhu, Roberto M Cesar Jr, Xiangyang Ji, and Xu-Cheng Yin. Dpflow: Adaptive optical flow estimation with a dual-pyramid framework. *arXiv preprint arXiv:2503.14880*, 2025.

Biswaroop Palit, Anup Basu, and Mrinal K Mandal. 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.

Jinyoung Park, Hee-Seon Kim, Kangwook Ko, Minbeom Kim, and Changick Kim. Videomamba: Spatio-temporal selective state space model. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024.

Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 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.

Bo Peng, Jianjun Lei, Huazhu Fu, Yalong Jia, Zongqian Zhang, and Yi Li. Deep video action clustering via spatio-temporal feature learning. *Neurocomputing*, 456:519–527, 2021.

Miao Qi, Ramzi Idoughi, and Wolfgang Heidrich. Hdnet: physics-inspired neural network for flow estimation based on helmholtz decomposition. *arXiv preprint arXiv:2406.08570*, 2024.

Zhaofan Qiu, Ting Yao, and Tao Mei. Learning deep spatio-temporal dependence for semantic video segmentation. *IEEE Transactions on Multimedia*, 20(4):939–949, 2017.

Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4161–4170, 2017.

Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. 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.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Tushar Sangam, Ishan Rajendrakumar Dave, Waqas Sultani, and Mubarak Shah. Transvisdrone: Spatio-temporal transformer for vision-based drone-to-drone detection in aerial videos. *arXiv preprint arXiv:2210.08423*, 2022.

Günter Schwarz. *Hodge Decomposition-A method for solving boundary value problems*. Springer, 2006.

Tobias Senst, Volker Eiselein, and Thomas Sikora. Robust local optical flow for feature tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(9):1377–1387, 2012.

Nusrat Sharmin and Remus Brad. Optimal filter estimation for lucas-kanade optical flow. *Sensors*, 12(9):12694–12709, 2012.

Chun-Chieh Shih, Hsiao-Rong Tyan, and HY Mark Liao. Shot change detection based on the reynolds transport theorem. In *Pacific-Rim Conference on Multimedia*, pp. 819–824. Springer, 2001.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Deqing Sun, Stefan Roth, and Michael J Black. 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.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 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.

Yu Sun, Xiyang Zhi, Haowen Han, Shikai Jiang, Tianjun Shi, Jinnan Gong, and Wei Zhang. Enhancing uav detection in surveillance camera videos through spatiotemporal information and optical flow. *Sensors*, 23(13):6037, 2023.

Michael Tao, Jiamin Bai, Pushmeet Kohli, and Sylvain Paris. Simpleflow: A non-iterative, sublinear optical flow algorithm. In *Computer graphics forum*, volume 31, pp. 345–353. Wiley Online Library, 2012.

Zachary Teed and Jia Deng. 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.

Tung Minh Tran, Doanh C Bui, Tam V Nguyen, and Khang Nguyen. 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.

Sareer Ul Amin, Bumsoo Kim, Yonghoon Jung, Sanghyun Seo, and Sangoh Park. Video anomaly detection utilizing efficient spatiotemporal feature fusion with 3d convolutions and long short-term memory modules. *Advanced Intelligent Systems*, 6(7):2300706, 2024.

Timo Von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. Human pose estimation from video and imus. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1533–1547, 2016.

Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pp. 36–54. Springer, 2024.

Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*, pp. 1385–1392, 2013.

F.M. White. *Fluid Mechanics*. McGraw-Hill series in mechanical engineering. McGraw Hill, 2011. ISBN 9780073529349. URL `https://books.google.com.au/books?id=egk8SQAACAAJ`.

Jian Wu, Bob Zhang, Yong Xu, and David Zhang. Illuminance compensation and texture enhancement via the hodge decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3):956–971, 2020.

Jonas Wulff and Michael J Black. 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.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017.

Jing Yuan, Gabriele Steidl, and Christoph Schnörr. Convex hodge decomposition of image flows. In *Joint Pattern Recognition Symposium*, pp. 416–425. Springer, 2008.

Jing Yuan, Christoph Schnörr, and Gabriele Steidl. Convex hodge decomposition and regularization of image flows. *Journal of Mathematical Imaging and Vision*, 33(2):169–177, 2009.

Christopher Zach, Thomas Pock, and Horst Bischof. 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.

Kaihao Zhang, Dongxu Li, Wenhan Luo, Wenqi Ren, and Wei Liu. Enhanced spatio-temporal interaction learning for video deraining: faster and better. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1287–1293, 2022.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pp. 766–782. Springer, 2016.

Tianyu Zhang, Longhui Wei, Lingxi Xie, Zijie Zhuang, Yongfei Zhang, Bo Li, and Qi Tian. Spatiotemporal transformer for video-based person re-identification. *arXiv preprint arXiv:2103.16469*, 2021.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 863–871, 2017.

Haidi Zhu, Haoran Wei, Baoqing Li, Xiaobing Yuan, and Nasser Kehtarnavaz. A review of video object detection: Datasets, metrics and methods. *Applied Sciences*, 10(21):7834, 2020.

Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 408–417, 2017.

Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. Towards high performance video object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7210–7218, 2018a.

Zhiqin Zhu, Hongpeng Yin, Yi Chai, Yanxia Li, and Guanqiu Qi. A novel multi-modality image fusion method based on image decomposition and sparse representation. *Information Sciences*, 432:516–529, 2018b.