Title

Texture-based optical flow for wind velocity estimation from water vapor data

Permalink

https://escholarship.org/uc/item/4zd3r3cx

Authors

Barnett, Joel R
Bertozzi, Andrea
Vese, Luminita A
et al.

Publication Date

2023-06-12

DOI

10.1117/12.2663008

Peer reviewed

Texture-based optical flow for wind velocity estimation from water vapor data

Joel R. Barnett[a], Andrea Bertozzi[a], Luminita A. Vese[a], and Igor Yanovsky[a]

[a]University of California, Los Angeles, 520 Portola Plaza, Los Angeles, CA, USA

ABSTRACT

Accurate estimation of atmospheric wind velocity plays an important role in weather forecasting, flight safety assessment and cyclone tracking. Atmospheric data captured by infrared and microwave satellite instruments provide global coverage for weather analysis. Extracting wind velocity fields from such data has traditionally been done through feature tracking, correlation/matching or optical flow means from computer vision. However, these recover either sparse velocity estimates, oversmooth details or are designed for quasi-rigid body motions which over-penalize vorticity and divergence within the often turbulent weather systems. We propose a texture based optical flow procedure tailored for water vapor data. Our method implements an $L1$ data term and total variation regularizer and employs a structure-texture image decomposition to identify key features which improve recoveries and help preserve the salient vorticity and divergence structures. We extend this procedure to a multi-fidelity scheme and test both flow estimation methods on simulated over-ocean mesoscale convective systems and convective and extratropical cyclone datasets, each of which have accompanying ground truth wind velocities so we can qualitatively compare performances with existing optical flow methods.

Keywords: Atmospheric wind estimation, optical flow, total variation

1. INTRODUCTION

Optical flow has a long and successful history in computer vision applications as a motion estimator. The method has found a place in object tracking including such applications as optical computer-mice, stereo-vision, and scene motion estimation. In general, an optical flow method attempts to extract a displacement field which describes the motion between two scenes (often two consecutive frames from an image sequence or video). While the method is traditionally used on natural imagery, it has been applied as a tool in studying fluids by examining particle-laden flows, a practice known as particle image velocimetry (PIV),[1,2] as well as for cloud and weather tracking.[3,4] Applications like PIV and cloud tracking prove challenging because of the complicated nature of fluid motion—in particular, there are no longer stable features or quasi-rigid motions typical of natural imagery.

We are interested in applying optical flow towards water vapor data with the aim of recovering atmospheric wind velocities. The ability to extract such estimates from satellite data is of significant interest since microwave and infrared instrument equipped satellites provide comprehensive coverage of weather phenomenon without the need for in-place instruments (e.g., ocean drifters, physical wind sensors), which are both costly and provide only sparse estimates. Our data comes from direct numerical simulations of atmospheric conditions from the Weather Research and Forecasting model,[5] and includes ground truth wind velocities for comparison. Water vapor from three weather scenarios is extracted from this nature-run data and used in our experiments. First, we have a mesoscale convective system (MCS) over the eastern Pacific which provides higher spatial resolution but with larger time intervals between frames. Second, we consider tropical convection (TC) over the maritime continent. Last, we consider an extratropical cyclone (ETC) event over the western Atlantic. The size, spatial and temporal resolution of each dataset is given in Table 1.

Table 1: Water vapor data description

Name	Image Size	Spatial Resolution	Time Resolution
MCS	850×1850 px^2	3000 m/px	900 s
TC	999×1299 px^2	3500 m/px	72 s
ETC	480×480 px^2	12000 m/px	120 s

2. BACKGROUND AND MODEL

The primary assumption of optical flow methods is one of *brightness constancy*. That is, the brightness of an object in a scene does not change over short time intervals. More precisely, let $I(\boldsymbol{x}, t) : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ represent the image intensity at some point $\boldsymbol{x} \in \Omega$ for each fixed time t, where $\boldsymbol{x} = (x, y)$. If we assume the object at $\boldsymbol{x}$ is displaced by $\boldsymbol{d} = \boldsymbol{d}(\boldsymbol{x}, t)$ as time advances to $t + \Delta t$, then brightness constancy assumes $I(\boldsymbol{x} + \boldsymbol{d}, t + \Delta t) = I(\boldsymbol{x}, t)$. Taylor expanding I about $(\boldsymbol{x}, t)$ gives the so-called optical flow constraint equation (OFCE)

$$I_t(\boldsymbol{x}, t) + \boldsymbol{u}^T \nabla I(\boldsymbol{x}, t) = 0, \tag{1}$$

where $\boldsymbol{u} = \boldsymbol{d}/\Delta t$ is the velocity field describing the displacement. By letting $\Delta t \to 0$, the left-hand side of (1) can also be seen as the total time-derivative of $I(\boldsymbol{x}, t)$, where it is understood that $(\boldsymbol{x}, t) = (\boldsymbol{x}(t), t)$ traces out the space-time path objects take as they move within the scene. Since (1) is obtained through linearization, it is sometimes called the linear OFCE. The non-linear version is simply

$$I(\boldsymbol{x} + \boldsymbol{d}, t + \Delta t) - I(\boldsymbol{x}, t) = 0. \tag{2}$$

In practice, when working with video or imagery data it is standard to take $\Delta t = 1$ (in which case $\boldsymbol{u} = \boldsymbol{d}$) and define frame one by $I_1(\boldsymbol{x}) = I(\boldsymbol{x}, t + 1)$ and frame zero by $I_0(\boldsymbol{x}) = I(\boldsymbol{x}, t)$. An optical flow method aims to find the flow field $\boldsymbol{u} = (u_1, u_2)^T$ given the images I_0 and I_1.

As it stands, (1) is ill-posed and additional constraints are required to determine the flow. This can be remedied by assuming the flow is constant on neighborhoods as was done by Lucas and Kanade,[6] or imposing regularization on the flow as was introduced by Horn and Schunck[7] (HS) in 1981. Methods which impose constraints on local regions, e.g. Lucas and Kanade's flow, block matching or correlation techniques,[8,9] are known as local methods which generally produce a sparse flow field describing the motion within a scene. Conversely, schemes which place requirements on the whole flow are known as global methods, and produce dense velocity fields. Additionally, hybrid methods[10–13] which blend correlation and varational techniques have been tested. Because we are interested in recovering dense flow estimates for wind velocities, we focus on global variational optical flow methods, which have outperformed cross correlation techniques for atmospheric tracking (see Ref. [3]).

Subsequent optical flow techniques have primarily been based on the seminal work of Horn and Schunck, which solves the minimization problem

$$\min_{\boldsymbol{u} = (u_1, u_2)} \lambda \int_\Omega \left(I_t + \boldsymbol{u}^T \nabla I \right)^2 d\boldsymbol{x} + \int_\Omega \left(|\nabla u_1|^2 + |\nabla u_2|^2 \right) d\boldsymbol{x}. \tag{3}$$

These have been applied to PIV,[1,2] tailored for fluid flow through relaxed vorticity and divergence penalties,[14] tested for weather forecasting[4] and utilized in atmospheric tracking.[3,15–19]

Each of these methods can broadly be categorized as the minimization problem

$$\min_{\boldsymbol{u}} \lambda \int_\Omega \psi(\rho(\boldsymbol{u}; I)) d\boldsymbol{x} + J(\boldsymbol{u}) \tag{4}$$

where ρ represents the data fitting term, ψ a non-negative penalizing function and $J(\cdot)$ is a regularization term. For instance, the HS flow is characterized by $\psi(x) = x^2$, $\rho(\boldsymbol{u}) = I_t + \boldsymbol{u}^T \nabla I$ and $J(\boldsymbol{u}) = \int_\Omega |\nabla u_1|^2 + |\nabla u_2|^2$. The positive constant λ weights the emphasis of the data over the regularization. For scenes with large displacements

between frames, the linearized OFCE is often inaccurate, so either non-linear schemes are devised (see Ref. 20 for e.g.), or a method of "warping", whereby $I(\boldsymbol{x} + \boldsymbol{u_0})$ is computed via interpolation (typically bicubic) for some approximation $\boldsymbol{u}_0$ to the desired flow. The OFCE is formed by linearizing (2) around the most recent approximation $\boldsymbol{x} + \boldsymbol{u}_0$. Additionally, to further assist with large displacements, optical flow algorithms commonly make use of a hierarchical pyramid scheme (see Ref. 21) in which spatially downsampled images are used to form flow estimates which then serve to initialize estimates for higher resolution images, iteratively repeating until the original resolution is reached (see Ref. 22 for a HS implementation).

2.1 Optical flow constraint equation limitations

Applying HS directly to fluid data yields less than satisfactory results. This is not surprising, as the HS method is designed for more rigid motions, with the regularizing term $\|\nabla \boldsymbol{u}\|^2$ penalizing turbulent flows (specifically the divergence and vorticity of $\boldsymbol{u}$) harshly.[14, 23] A natural starting place to try and improve flow estimates is to modify $\psi(\rho)$ and J to be more suited for the data. A more robust data term such as $L1$, which the authors in Refs. 24–26 consider, in addition to total variation regularization, better adapt to illumination changes and promote discontinuities common in complicated flows. Efficient implementations of this so named TV-$L1$ optical flow method are discussed in Refs. 24–26, along with a non-linear scheme which requires no warping in Ref. 20. A Python language implementation is available from the Scikit-image[27] library. The method has also been proposed for atmospheric motion vector estimation, and the authors in Refs. 16, 17 show that TV-$L1$ performs favorably compared with conventional atmospheric motion tracking methods. TV-$L1$ improves flow estimates in our data compared with HS and is able to extract the general behavior of the flow, but struggles to recover flow in areas of low image variation and over-smooths the divergence and vorticity present in the flow.

While additionally modifying the penalty ψ and regularization J may yield some improvements, we consider more generally whether the OFCE alone is appropriate for recovering the desired flow. Because we have the ground truth data $\boldsymbol{u}_{GT}$ available, we are able to test the performance of the ground truth and any computed flow $\boldsymbol{u}$ at minimizing (4) as well as satisfying the data term $\rho(\cdot; I)$. We make the critical observation that the TV-$L1$ computed flows outperform the ground truth, both at minimizing (4) and satisfying the OFCE (both linear and non-linear adaptations), yet do not match the desired flow satisfactorily. This indicates that for our data, the OFCE alone is insufficient for guiding flow discovery. Driving $\boldsymbol{u}$ to satisfy the OFCE more exactly can move $\boldsymbol{u}$ further from $\boldsymbol{u}_{GT}$, and additions or modifications to the constraint equation are needed if we hope to recover improved estimates.

2.2 Texture features for flow estimation

In an attempt to provide more context to the flow estimator in regions of uniform image-intensity, we propose decomposing the image into texture and structure components, thereafter extracting a velocity field from a combination of these components. We note that using the texture portion of images for optical flow has been proposed as a way to combat the effects of brightness variation on flow estimation (i.e. illumination changes which break the brightness constancy assumption),[25] however in the context of water-vapor imagery, the data is not illumination based and so our motivation for such a decomposition is distinct.

A standard image decomposition can be formed by the iconic Rudin-Osher-Fatemi total variation denoising model, where an image I is split into the sum $I = I^S + I^{Text}$, with the structure portion I^S minimizing

$$\min_{f \in BV(\Omega)} \lambda \int_{\Omega} (I - f)^2 d\boldsymbol{x} + TV(f), \tag{5}$$

where $BV(\Omega)$ is the space of functions of bounded variation over Ω. However, a decomposition with more oscillatory texture is desirable because this provides more features to add context for flow extraction, especially in regions of low image variation. One such texture-structure image decomposition was proposed by Meyer,[28] in which $I^{Text} = I - I^S$ belongs to the weaker space $G(\Omega) = \{\text{div}(\boldsymbol{g}) : \boldsymbol{g} \in L^{\infty}(\Omega)\}$. Oscillations such as texture have small norms in $G(\Omega)$, and thus are not penalized by such a model. The decomposition is determined by solving

$$I^S = \underset{f \in BV(\Omega), \boldsymbol{g} \in L^{\infty}(\Omega)}{\arg\min} \lambda \int_{\Omega} (I - (f + \text{div}(\boldsymbol{g}))^2 d\boldsymbol{x} + TV(f) + \mu \|\boldsymbol{g}\|_{\infty}. \tag{6}$$

Details on numerically determining (6) are found in Ref. 29, that we utilize to find $I^{Text} = I - I^S$ for our data.

2.2.1 Texture-flow

Armed with I^{Text} computed for both frames I_0 and I_1, we compute the TV-$L1$ texture-flow $\boldsymbol{u}^{Text}$ determined by these textural components (i.e. solve (4) with $\psi(x) = |x|$, $\rho(\boldsymbol{u}; I^{Text}) = I_1^{Text}(\boldsymbol{x} + \boldsymbol{u}) - I_0^{Text}(\boldsymbol{x})$ and $J(\boldsymbol{u}) = TV(u_1) + TV(u_2)$). This "texture-based" flow can be further improved by finding the flow refinement $\boldsymbol{u}^R$ which further registers (according to $\psi(\rho(\cdot))$) the warped image $I_1(\boldsymbol{x} + \boldsymbol{u}^{Text})$ to I_0, producing a total flow $\boldsymbol{u} = \boldsymbol{u}^{Text} + \boldsymbol{u}^R$. This texture-flow process is encoded in (7)

$$
\begin{cases}
\boldsymbol{u}^{Text} & \leftarrow \arg\min_{\boldsymbol{u}} \lambda \int_{\Omega} \psi(\rho(\boldsymbol{u}; I^{Text})) d\boldsymbol{x} + J(\boldsymbol{u}) \\
I_1 & \leftarrow I_1(\boldsymbol{x} + \boldsymbol{u}^{Text}) \\
\boldsymbol{u}^R & \leftarrow \arg\min_{\boldsymbol{u}} \lambda \int_{\Omega} \psi(\rho(\boldsymbol{u}; I)) d\boldsymbol{x} + J(\boldsymbol{u}) \\
\boldsymbol{u} & \leftarrow \boldsymbol{u}^{Text} + \boldsymbol{u}^R.
\end{cases}
\tag{7}
$$

The hope is that the texture within I_0^{Text} and I_1^{Text} affords enough information to determine $\boldsymbol{u}^{Text}$ more accurately in regions of uniform image intensity, while $\boldsymbol{u}^R$ corrects for the motions of the larger structures present within I_0 and I_1.

2.2.2 Multi-fidelity flow

Alternatively, one can reconstruct the full flow in a single process by concurrently considering the original and textural portions of the data I_0 and I_1. This is done by creating a data fidelity term which incorporates multiple features—in this case, the raw data I_0, I_1 and it's texture I_0^{Text}, I_1^{Text}—from the data simultaneously, as given in the following multi-fidelity minimization problem

$$
\arg\min_{\boldsymbol{u}} \lambda_1 \int_{\Omega} |\rho(\boldsymbol{u}; I)| + \lambda_2 \int_{\Omega} |\rho(\boldsymbol{u}; I^{Text})| d\boldsymbol{x} + TV(u_1) + TV(u_2),
\tag{8}
$$

where the λ_i are weights for each fidelity term.

3. NUMERICAL MINIMIZATION

To solve for the flow, we must minimize (4) for a specific choice of ψ, ρ and J. In TV-$L1$ flow, the corresponding problem is

$$
\min_{\boldsymbol{u}} \lambda \int_{\Omega} |\rho(\boldsymbol{u}; I)| d\boldsymbol{x} + TV(u_1) + TV(u_2),
$$

where $\rho(\boldsymbol{u}; I) = I_1(\boldsymbol{x} + \boldsymbol{u}) - I_0(\boldsymbol{x})$ is highly nonlinear. To approach this minimization, we linearize ρ about a current flow estimate $\boldsymbol{u}_0$,

$$
\rho(\boldsymbol{u}; I) \approx \rho(\boldsymbol{u}; \boldsymbol{u}_0, I) := I_1(\boldsymbol{x} + \boldsymbol{u}_0) - I_0(\boldsymbol{x}) + (\boldsymbol{u} - \boldsymbol{u}_0)^T \nabla I(\boldsymbol{x} + \boldsymbol{u}_0),
$$

where, with some abuse of notation we refer to the linearized version by $\rho(\boldsymbol{u}; \boldsymbol{u}_0, I)$. At times when the context is clear, we will drop the explicit dependence of ρ on I. We approach minimizing

$$
\min_{\boldsymbol{u}} \lambda \int_{\Omega} |I_1(\boldsymbol{x} + \boldsymbol{u}_0) - I_0(\boldsymbol{x}) + (\boldsymbol{u} - \boldsymbol{u}_0)^T \nabla I(\boldsymbol{x} + \boldsymbol{u}_0)| d\boldsymbol{x} + TV(u_1) + TV(u_2)
\tag{9}
$$

by splitting the fidelity and regularizing terms, alternatingly solving the two proximal problems (10) and (11)

$$
\boldsymbol{u}^{k+1} \in \arg\min_{\boldsymbol{u}} \lambda \int_{\Omega} |\rho(\boldsymbol{u}; \boldsymbol{u}_0)| d\boldsymbol{x} + \frac{1}{2\theta} \|\boldsymbol{u} - \boldsymbol{v}^k\|^2
\tag{10}
$$

$$
\boldsymbol{v}^{k+1} \in \arg\min_{\boldsymbol{v}} \frac{1}{2\theta} \|\boldsymbol{u}^{k+1} - \boldsymbol{v}\|^2 + J(\boldsymbol{v}),
\tag{11}
$$

with $\boldsymbol{u}^0$ initialized to some guess for the flow (typically the current estimate $\boldsymbol{u}_0$). The tightness parameter θ forces $\boldsymbol{v}^k$ near $\boldsymbol{u}^k$, and letting $\theta \to 0$ the split minimization problems given in (10) and (11) is equivalent with (9). The reason for splitting is that the individual minimization problems are easier when considered separately.

Indeed, (11) decouples along the components of $\boldsymbol{v}$, giving two ROF total variation denoising problems. These have a well documented (see Ref. 30) fast dual-projection method,

$$\boldsymbol{p}_i^{n+1} = \frac{\boldsymbol{p}_i^n + \tau \nabla(\mathrm{div}\boldsymbol{p}_i^n - u_i^k/\theta)}{1 + \tau|\nabla(\mathrm{div}\boldsymbol{p}_i^n - u_i^k)|} \tag{12}$$

$$v_i^{k+1} = u_i^k - \theta \mathrm{div}\boldsymbol{p}_i^N, \tag{13}$$

which can be used to solve for each component, v_i^{k+1}, of $\boldsymbol{v}^{k+1}$ after running n in (12) to some convergence criterion N. For (10), we follow a shrinkage-like thresholding procedure (14) detailed in Refs. 24–26

$$\boldsymbol{u}^{k+1} = \boldsymbol{v}^k + \begin{cases} \lambda\theta\nabla I_1(\boldsymbol{x} + \boldsymbol{u}_0) & \text{if } \rho(\boldsymbol{v}^k; \boldsymbol{u}_0) < -\lambda\theta|\nabla I_1(\boldsymbol{x} + \boldsymbol{u}_0)|^2 \\ -\lambda\theta\nabla I_1(\boldsymbol{x} + \boldsymbol{u}_0) & \text{if } \rho(\boldsymbol{v}^k; \boldsymbol{u}_0) > \lambda\theta|\nabla I_1(\boldsymbol{x} + \boldsymbol{u}_0)|^2 \\ -\rho(\boldsymbol{v}^k; \boldsymbol{u}_0)\nabla I_1(\boldsymbol{x} + \boldsymbol{u}_0)/|\nabla I_1(\boldsymbol{x} + \boldsymbol{u}_0)|^2 & \text{if } |\rho(\boldsymbol{v}^k; \boldsymbol{u}_0)| \le \lambda\theta|\nabla I_1(\boldsymbol{x} + \boldsymbol{u}_0)|^2. \end{cases} \tag{14}$$

This process which solves (9) can be used in both the minimization steps in the *texture-flow* procedure (7). We use the Python TV-L1 implementation available from the Scikit-image image processing library[27] when solving (9).

The splitting strategy has the additional benefit of simplifying the process of adding or removing terms from the fidelity (see Ref. 18 for further exploration), and we can easily test the influence of including a texture-based fidelity term in the *multi-fidelity* problem (8). We are able to solve problems in the form of (10) and (11), so our goal is to use splitting to recast (8) into these forms. The constrained optimization,

$$\begin{cases} \arg\min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}} \lambda_1 \int_\Omega |\rho(\boldsymbol{u}; \boldsymbol{u}_0, I)|d\boldsymbol{x} + \lambda_2 \int_\Omega |\rho(\boldsymbol{v}; \boldsymbol{u}_0, I^{Text})|d\boldsymbol{x} + J(\boldsymbol{w}) \\ \text{subject to} \quad \boldsymbol{u} = \boldsymbol{v} = \boldsymbol{w}, \end{cases} \tag{15}$$

is equivalent with (8) and we can relax (15) by adding the quadratic penalty

$$Q(x_1, x_2, \dots, x_n, \theta) = \frac{1}{2\theta} \sum_{i \ne j} \|x_i - x_j\|^2,$$

giving the unconstrained problem

$$\arg\min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}} \lambda_1 \int_\Omega |\rho(\boldsymbol{u}; \boldsymbol{u}_0, I)|d\boldsymbol{x} + \lambda_2 \int_\Omega |\rho(\boldsymbol{v}; \boldsymbol{u}_0, I^{Text})|d\boldsymbol{x} + J(\boldsymbol{w}) + Q(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}, \theta). \tag{16}$$

The parameter $\theta > 0$, and for $\theta \downarrow 0$, (16) is equivalent with (8) and (15). We will approach (16) by alternatingly minimizing in $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ and noting that

$$Q(x, y, z, \theta) = \frac{1}{2\theta}\left(\|x - y\|^2 + \|x - z\|^2 + \|y - z\|^2\right)$$
$$= \frac{1}{\theta}\left(\left\|x - \frac{y + z}{2}\right\|^2 + c(y, z)\right),$$

where c is some function independent of x. Since Q is symmetric about all but its last argument, we can equivalently reduce the dependence of Q on y and z to the same quadratic penalty. Our iterative scheme is

$$\boldsymbol{u}^{k+1} \in \arg\min_{\boldsymbol{u}} \lambda_1 \int_\Omega |\rho(\boldsymbol{u}; \boldsymbol{u}_0, I)|d\boldsymbol{x} + \frac{1}{\theta}\left\|\boldsymbol{u} - \frac{\boldsymbol{v}^k + \boldsymbol{w}^k}{2}\right\|^2 \tag{17}$$

$$\boldsymbol{v}^{k+1} \in \arg\min_{\boldsymbol{v}} \lambda_2 \int_\Omega |\rho(\boldsymbol{v}; \boldsymbol{u}_0, I^{Text})|d\boldsymbol{x} + \frac{1}{\theta}\left\|\boldsymbol{v} - \frac{\boldsymbol{u}^{k+1} + \boldsymbol{w}^k}{2}\right\|^2 \tag{18}$$

$$\boldsymbol{w}^{k+1} \in \arg\min_{\boldsymbol{w}} J(\boldsymbol{w}) + \frac{1}{\theta}\left\|\boldsymbol{w} - \frac{\boldsymbol{u}^{k+1} + \boldsymbol{v}^{k+1}}{2}\right\|^2, \tag{19}$$

starting with some initialization for $\boldsymbol{u}^0$, $\boldsymbol{v}^0$ and $\boldsymbol{w}^0$ (usually the current flow estimate $\boldsymbol{u}_0$). Again, the problem has been reduced into proximal problems and an ROF problem, which can be solved via the explicit formula (14) and dual projection procedure (12).

Computing the derivative terms in these schemes is done with finite differences. Central differences are used in the gradient terms in (14) and forward and backward differences are composed when computing the gradient-of-divergence terms in (12).

In order to overcome inaccuracies from large displacements (in terms of pixels), we also incorporate a hierarchical pyramid scheme (see Ref. 21), downsampling I_1 and I_0 until the resolution is reduced sufficiently so that the displacements no longer span many gridpoints. Then, recoveries are determined at each resolution level, starting from the coarsest and working towards the original resolution, with each recovery acting as an initialization for the next pyramid level. The full algorithm with warping for the multi-fidelity flow (8) is given below.

Algorithm 1 Multi-fidelity Optical Flow

Input: images I_0, I_1, I_0^{Text}, I_1^{Text}, initial flow estimate $\boldsymbol{u}_0$, parameters N_{warp}, N_{iter}, N_{levels}, θ, and λ_1, λ_2.
Output: flow estimate $\boldsymbol{u}$

1: Form downsampled pyramid levels $I_{0,l}$, $I_{1,l}$, $\boldsymbol{u}_{0,l}$, $l = 1, 2, \ldots, N_{levels}$.
2: **for** $l = N_{levels}, \ldots, 2, 1$ **do**
3:     **for** $k = 0, 1, \ldots, N_{warp} - 1$ **do**
4:         Compute $I_{1,l}(\boldsymbol{x} + \boldsymbol{u}_{k,l})$, $\nabla I_{1,l}(\boldsymbol{x} + \boldsymbol{u}_{k,l})$, $I_{1,l}^{Text}(\boldsymbol{x} + \boldsymbol{u}_{k,l})$ via bicubic uplook
5:         $\boldsymbol{u}^0, \boldsymbol{v}^0, \boldsymbol{w}^0 \leftarrow \boldsymbol{u}_{k,l}$
6:         **for** $n = 0, 1, \ldots, N_{iter} - 1$ **do**
7:

$$\boldsymbol{u}^{n+1} \leftarrow \underset{\boldsymbol{u}}{\arg\min} \; \lambda_1 \int_{\Omega} |\rho(\boldsymbol{u}; \boldsymbol{u}_{k,l}, I_l)| d\boldsymbol{x} + \frac{1}{\theta} \left\| \boldsymbol{u} - \frac{\boldsymbol{v}^n + \boldsymbol{w}^n}{2} \right\|^2$$

$$\boldsymbol{v}^{n+1} \leftarrow \underset{\boldsymbol{v}}{\arg\min} \; \lambda_2 \int_{\Omega} |\rho(\boldsymbol{u}; \boldsymbol{u}_{k,l}, I_l^{Text})| d\boldsymbol{x} + \frac{1}{\theta} \left\| \boldsymbol{v} - \frac{\boldsymbol{u}^{n+1} + \boldsymbol{w}^n}{2} \right\|^2$$

$$\boldsymbol{w}^{n+1} \leftarrow \underset{\boldsymbol{w}}{\arg\min} \; \frac{1}{\theta} \left\| \boldsymbol{w} - \frac{\boldsymbol{u}^{n+1} + \boldsymbol{v}^{n+1}}{2} \right\|^2 + J(\boldsymbol{w})$$

8:         **end for**
9:         $\boldsymbol{u}_{k+1,l} \leftarrow \boldsymbol{w}^{N_{iter}}$
10:     **end for**
11:     $\boldsymbol{u}_{0,l+1} \leftarrow \text{Upsample}(\boldsymbol{u}_{N_{warp},l})$
12: **end for**
13: Return $\boldsymbol{u}_{N_{warp}, N_{levels}}$

---

4. RESULTS

In this section we discuss the results of our texture-based and multi-fidelity flows and compare them against existing optical flow methods. In particular, we will compare against the classic HS optical flow method as well as the popular TV-$L1$ scheme, which recently has shown success[16,17] over the conventional feature-tracking algorithm which is widely used, including for the National Oceanic and Atmospheric Administration (NOAA) Geostationary Operational Environmental Satellites (GOES) cloud motion retrieval.

Recall we are interested in determining a vector field describing the atmospheric wind velocity given a sequence of water-vapor images. A typical water vapor image is given in Fig. 1. In all our experiments, we take in two images to guide flow retrieval which are separated by $2\Delta t$, where Δt is the time between frames in the image sequence. The wind velocity associated with the intermediate frame is taken as the ground truth, and the
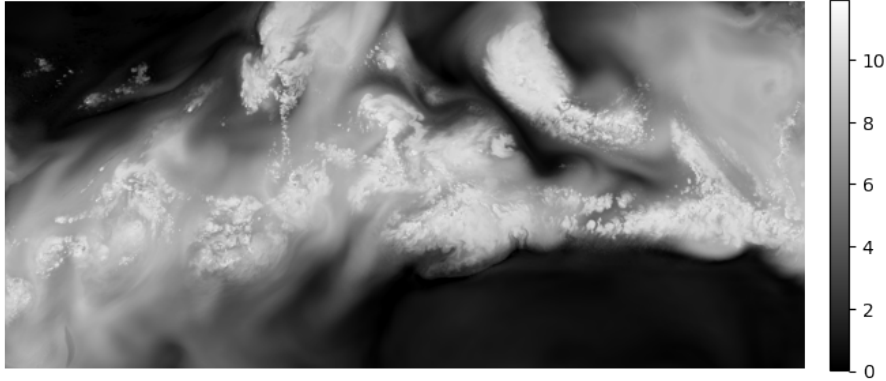
Figure 1: A typical water-vapor image, in this case from the MCS. Pixel values correspond to the water vapor mixing ratio in g/kg.
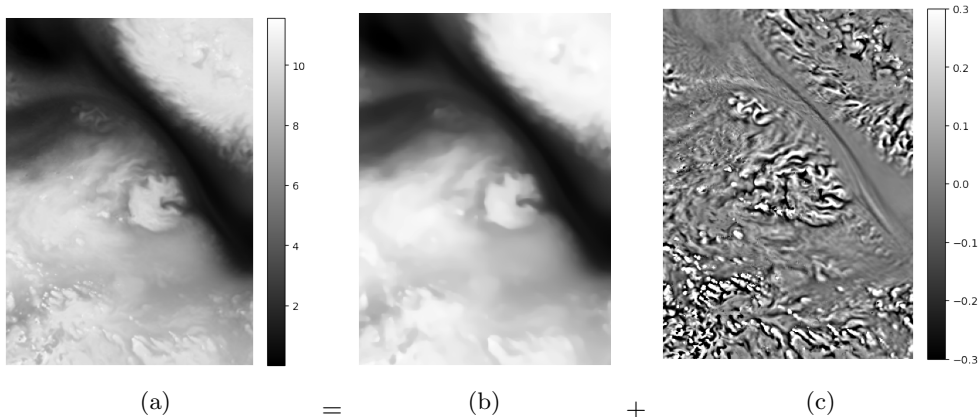

(a)   =   (b)   +   (c)

Figure 2: Structure-texture decomposition of a cropped portion of Fig. 1. The original image I is given in (a), it's structure portion I^S in (b), and textural portion I^{Text} in (c). Recall $I = I^S + I^{Text}$.

estimated vector field is compared against the ground truth using root-mean-squared-vector-distance (RMSVD). The RMSVD between two vector fields $\boldsymbol{u}, \boldsymbol{v}$ defined on an $M \times N$ pixel grid is

$$RMSVD(\boldsymbol{u}, \boldsymbol{v}) = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (v_{1,ij} - u_{1,ij})^2 + (v_{2,ij} - u_{2,ij})^2}.$$

We also make use of a hierarchical pyramid scheme for each method, downsampling by a factor of 2 for 10 levels (or until the minimum dimension of the image is 16 pixels).

As discussed in Section 2, using optical flow for wind velocity estimation can be challenging for a variety of reasons. We are considering several datasets, and the ground truth displacements between frames can be large (in terms of pixels), violating the linearization assumptions when forming the OFCE. On the other hand, in the case of very coarse spatial resolution such as with the ETC dataset, the displacements can be quite small, so that the ground truth motion vectors are at a subpixel level. These issues are handled by a combination of the hierarchical pyramid and warping routines. However, atmospheric motion is complex and turbulent, and large regions of uniform water vapor within the data provide little information for a flow estimator to extract a velocity estimate accurately.

This last issue is the primary motivation for incorporating image texture into the estimation procedure. In our texture-flow scheme, we decompose an image into its structural and textural portions, as shown for a crop of the MCS dataset in Fig. 2, and use the textured portion for motion estimation. In Fig. 3, we visualize the wind velocity field using a colormap, where color indicates the direction of the wind and saturation indicates

Table 2: RMSVD in m/s and px/Δt between flow estimations and ground truth wind velocity. Here, Δt is the time between frames for each respective dataset. Bold entries indicate lowest error in each row.

	HS		TVL1		Texture-flow		Multi-fidelity	
	m/s	px/Δt	m/s	px/Δt	m/s	px/Δt	m/s	px/Δt
MCS	3.400	1.020	2.483	0.745	**1.985**	**0.596**	**1.985**	**0.596**
TC	2.903	0.060	2.462	0.051	2.361	0.049	**2.299**	**0.047**
ETC	6.205	0.062	5.855	0.059	5.838	0.058	**5.675**	**0.057**

the speed. Our tests extracting wind velocity estimates using the textured portion, I^{Text}, of an image alone demonstrate that texture is an excellent feature for atmospheric motion tracking. This is made clear in Fig. 3d when compared with 3c. Fig. 3d is obtained from only the information contained in I^{Text}, while 3c is formed from the whole image, yet 3d outperforms the TV-$L1$ estimate 3c. Of note, we see sizable improvement within the boxed regions of 3d, which represent portions of I with relatively uniform water-vapor levels (see Fig. 1). This indicates that texture indeed provides good context within these challenging regions, and our motivation for including texture in the flow retrieval process is valid. The remaining images in Fig. 3 show the complete texture-flow and multi-fidelity flow estimates, along with the HS flow for comparison.

Similar flow visualizations are given for the TC and ETC datasets in Figures 4 and 5. The texture-flow and multi-fidelity flow produce 8–40% improved flow estimates compared with HS and 3–20% improvements over the TV-$L1$ methods across the three datasets. Full results are recorded in Table 2. We remark that lower performance in units m/s does not necessarily imply lower performance relative to the resolution of the data collected. For instance, since the ETC dataset has a very coarse spatial resolution (12,000m per pixel) and moderate temporal resolution ($\Delta t = 120$s), an error of 1 m/s corresponds to 0.01 px/Δt.

5. CONCLUSION

In this work we demonstrate that optical flow can effectively perform wind velocity estimation from remote sensing water-vapor data. Additionally, we show that texture is a pertinent feature for motion estimation and we introduce two texture-based optical flow procedures.

The first method, "texture-flow", arrived from observing that the optical flow constraint equation (OFCE) on which the vast majority of variational optical flow methods are built, cannot recover the desired complex flow of atmospheric motion alone. Modifying the regularization term or considering a non-linear OFCE approach is insufficient, and modified or additional data fidelity terms should be considered. We determine that image-texture is a viable fidelity term, and augment the fidelity with a textural portion of the data.

The second "multi-fidelity" method extends the idea from texture-flow by developing a convenient algorithm which has the advantage of easily handling additional fidelity terms, and can be extended to include many more image-features in the flow retrieval process. In future works, we would like to test additional image-features including non-local methods and develop stopping criteria for strategically halting the process.

Collectively, these methods perform well on convective and cylconic weather systems, and this research would benefit the microwave and infrared instrument systems that commonly record such data. Both methods outperform the Horn Schunck and TV-$L1$ schemes across varying spatial and temporal resolutions.

ACKNOWLEDGMENTS

(a) Ground truth flow

(b) HS flow RMSVD = 3.400 m/s

(c) TVL1 flow: RMSVD = 2.483 m/s.

(d) Flow from I^{Text} alone: RMSVD = 1.996 m/s.

(e) Texture-flow from (7): RMSVD = 1.985 m/s.

(f) Multi-fidelity flow from (8): RMSVD = 1.985 m/s.

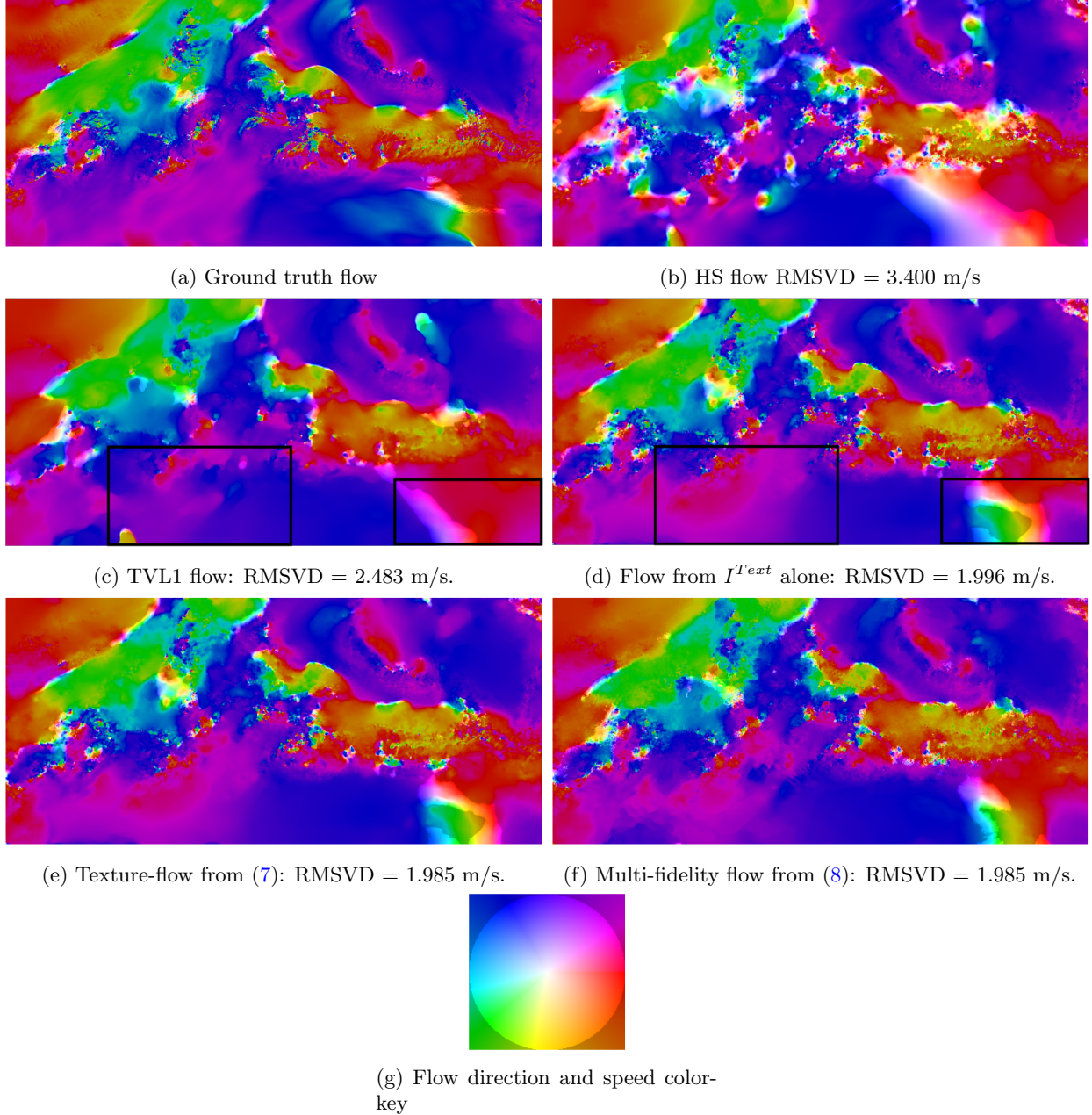(g) Flow direction and speed color-key

Figure 3: Colored flow visualization for a mesoscale convective system (MCS). The boxed regions in (c) and (d) indicate areas where the image has primarily uniform water-vapor levels, and show the improvement that the texture features add to velocity estimation. In (g), the color-key color indicates velocity direction and the saturation indicates flow velocity magnitude.

REFERENCES

[1] Adrian, R. J., "Twenty years of particle image velocimetry," *Experiments in Fluids* **39**(2), 159–169 (2005).

[2] Lu, J., Yang, H., Zhang, Q., and Yin, Z., "An accurate optical flow estimation of piv using fluid velocity decomposition," *Experiments in Fluids* **62**(4), 1–16 (2021).

[3] Chow, C. W., Belongie, S., and Kleissl, J., "Cloud motion and stability estimation for intra-hour solar forecasting," *Solar Energy* **115**, 645–655 (2015).
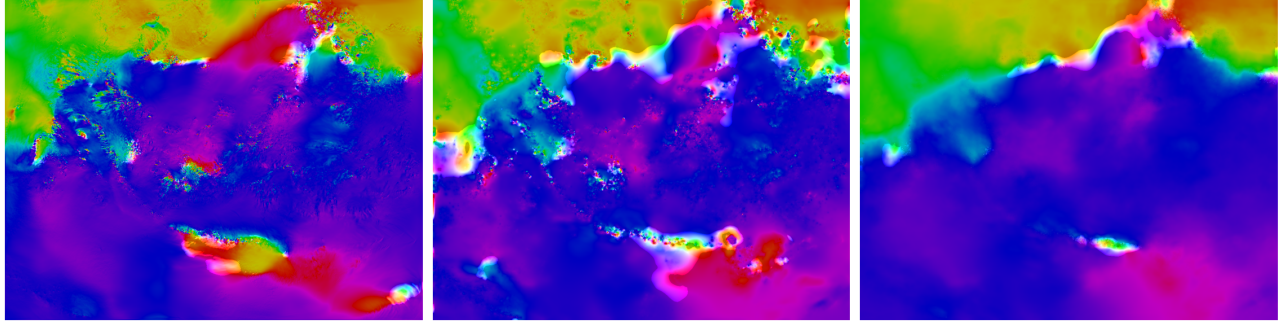
(a) Ground truth flow   (b) HS flow RMSVD = 2.903 m/s   (c) TVL1 flow: RMSVD = 2.462 m/s.
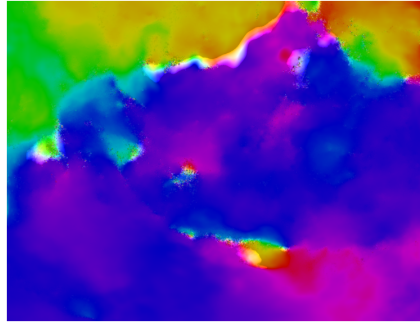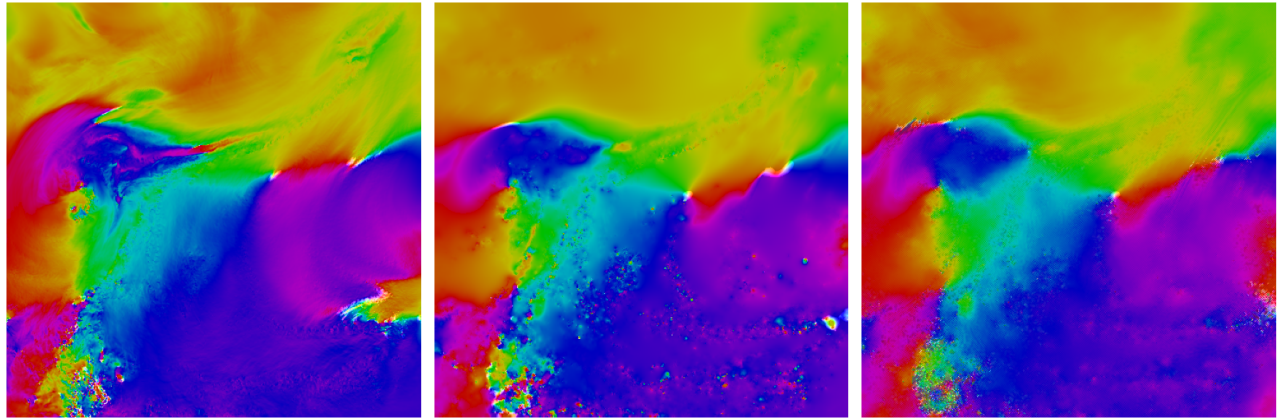
(d) Texture-flow from (7): RMSVD = 2.361 m/s.
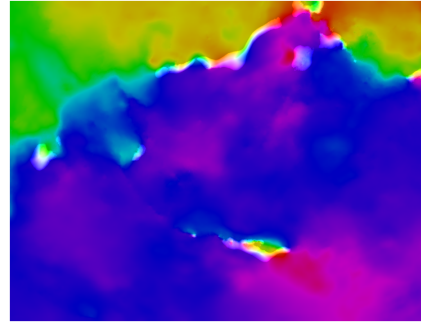
(e) Multi-fidelity flow from (8): RMSVD = 2.299 m/s.

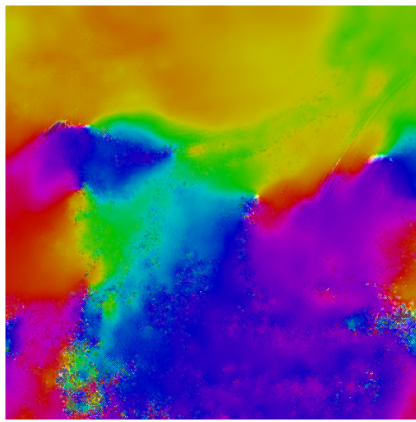Figure 4: Colored flow visualization for tropical convection (TC) data.

[4] Woo, W.-c. and Wong, W.-k., "Operational application of optical flow techniques to radar-based rainfall nowcasting," *Atmosphere* **8**(3) (2017).

[5] Skamarock, C., Klemp, B., Dudhia, J., Gill, O., Barker, D. M., Duda, G., Huang, X., Wang, W., and Powers, G., "A description of the advanced research wrf version 3," U.S. National Center for Atmospheric Research (2008).

[6] Lucas, B. D., Kanade, T., et al., [*An iterative image registration technique with an application to stereo vision*], vol. 81, Vancouver (1981).

[7] Horn, B. K. and Schunck, B. G., "Determining optical flow," *Artificial intelligence* **17**(1-3), 185–203 (1981).

[8] Liu, T., Merat, A., Makhmalbaf, M. H. M., Fajardo, C., and Merati, P., "Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images," *Experiments in Fluids* **56**(8), 166 (2015).

[9] Posselt, D. J., Wu, L., Mueller, K., Huang, L., Irion, F. W., Brown, S., Su, H., Santek, D., and Velden, C. S., "Quantitative assessment of state-dependent atmospheric motion vector uncertainties," *Journal of Applied Meteorology and Climatology* **58**(11), 2479–2495 (2019).

[10] Heitz, D., Héas, P., Mémin, E., and Carlier, J., "Dynamic consistent correlation-variational approach for robust optical flow estimation," *Experiments in Fluids* **45**(4), 595–608 (2008).

[11] Bruhn, A., Weickert, J., and Schnörr, C., "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International journal of computer vision* **61**, 211–231 (2005).

[12] Heitz, D., Mémin, E., and Schnörr, C., "Variational fluid flow measurements from image sequences: synopsis and perspectives," *Experiments in fluids* **48**(3), 369–393 (2010).

[13] Liu, T., Salazar, D. M., Fagehi, H., Ghazwani, H., Montefort, J., and Merati, P., "Hybrid optical-flow-cross-correlation method for particle image velocimetry," *Journal of Fluids Engineering* **142**(5) (2020).

[14] Chen, X., Zillé, P., Shao, L., and Corpetti, T., "Optical flow for incompressible turbulence motion estimation," *Experiments in fluids* **56**(1), 1–14 (2015).
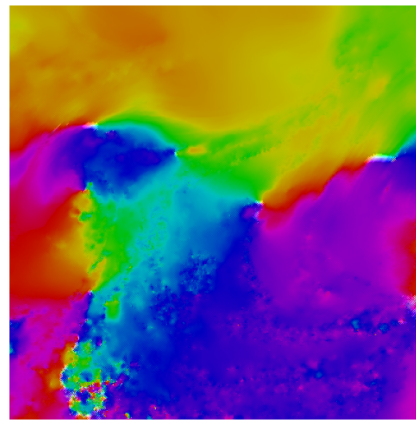
(a) Ground truth flow   (b) HS flow RMSVD = 6.205 m/s   (c) TVL1 flow: RMSVD = 5.855 m/s.

(d) Texture-flow from (7): RMSVD = 5.838 m/s.

(e) Multi-fidelity flow from (8): RMSVD = 5.675 m/s.

Figure 5: Colored flow visualization of extratropical cylcone (ETC) data.

[15] Bresky, W. and Daniels, J., "The feasibility of an optical flow algorithm for estimating atmospheric motion," in [*Proceedings of the Eighth Int. Winds Workshop, Beijing, China*], 24–28, Citeseer (2006).

[16] Yanovsky, I., Su, H., Posselt, D. J., Hristova-Veleva, S., Wu, L., Lambrigtsen, B., and Nguyen, H., "Atmospheric motion vector retrieval using the optical flow method," in [*25th Conference on Satellite Meteorology, Oceanography, and Climatology, Joint 2022 NOAA Satellite Conference*], AMS Collective Madison Meeting (2022).

[17] Yanovsky, I., Posselt, D. J., Wu, L., Hristova-Veleva, S., Nguyen, H., Lambrigtsen, B., and Zeng, X., "Atmospheric motion vector retrieval using the total variation-based optical flow method," IEEE Geoscience and Remote Sensing Symposium (2023).

[18] Barnett, J. R., Bertozzi, A., Vese, L. A., and Yanovsky, I., "Incorporating texture features into optical flow for atmospheric wind velocity estimation," IEEE Geoscience and Remote Sensing Symposium (2023).

[19] Ouyed, A., Zeng, X., Wu, L., Posselt, D., and Su, H., "Two-stage artificial intelligence algorithm for calculating moisture-tracking atmospheric motion vectors," *Journal of Applied Meteorology and Climatology* **60**(12), 1671 – 1684 (2021).

[20] Steinbrücker, F., Pock, T., and Cremers, D., "Large displacement optical flow computation withoutwarping," in [*2009 IEEE 12th International Conference on Computer Vision*], 1609–1614, IEEE (2009).

[21] Anandan, P., Bergen, J. R., Hanna, K. J., and Hingorani, R., "Hierarchical model-based motion estimation," in [*Motion analysis and image sequence processing*], 1–22, Springer (1993).

[22] Meinhardt-Llopis, E. and Sánchez, J., "Horn-schunck optical flow with a multi-scale strategy," *Image Processing on line* (2013).

[23] Corpetti, T., Heitz, D., Arroyo, G., Mémin, E., and Santa-Cruz, A., "Fluid experimental flow estimation based on an optical-flow scheme," *Experiments in fluids* **40**(1), 80–97 (2006).

[24] Sánchez Pérez, J., Meinhardt-Llopis, E., and Facciolo, G., "TV-L1 Optical Flow Estimation," *Image Processing On Line* **3**, 137–150 (2013).

[25] Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D., "An improved algorithm for tv-l 1 optical flow," in [*Statistical and geometrical approaches to visual motion analysis*], 23–45, Springer (2009).

[26] Zach, C., Pock, T., and Bischof, H., "A duality based approach for realtime tv-l 1 optical flow," in [*Joint pattern recognition symposium*], 214–223, Springer (2007).

[27] van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors, "scikit-image: image processing in python," *PeerJ* **2**, e453 (jun 2014).

[28] Meyer, Y., [*Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*], vol. 22, American Mathematical Soc. (2001).

[29] Osher, S., Solé, A., and Vese, L., "Image decomposition and restoration using total variation minimization and the h^{-1} norm," *Multiscale Modeling & Simulation* **1**(3), 349–370 (2003).

[30] Chambolle, A., "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision* **20**(1), 89–97 (2004).