A Higher-Order Model for Fluid Motion Estimation

Wei Liu and Eraldo Ribeiro

Computer Vision and Bio-Inspired Computing Laboratory,
Florida Institute of Technology, Melbourne, FL 32901, USA
lwei@my.fit.edu, eribeiro@cs.fit.edu,

Abstract. Image-based fluid motion estimation is of interest to science and engineering. Flow-estimation methods often rely on physics-based or spline-based parametric models, as well as on smoothing regularizers. The calculation of physics models can be involved, and commonly used 2nd-order regularizers can be biased towards lower-order flow fields. In this paper, we propose a local parametric model based on a linear combination of complex-domain basis flows, and a resulting global field that is produced by blending together local models using partition-of-unity. We show that the global field can be regularized to an *arbitrary order* without bias towards specific flows. Additionally, the blending approach to fluid-motion estimation is more flexible than competing spline-based methods. We obtained promising results on both synthetic and real fluid data.

Keywords: Fluid-flow estimation, optical flow, holomorphic functions.

1 Introduction

Estimating fluid motion from images is interesting to many science and engineering applications, and has received renewed attention from the computer vision community [2,7]. Fluid-flow estimation differs from the similar optical-flow estimation problem in a number of ways [5]. First, general optical flow fields are often unstructured, while fluid flows usually result from continuous physical processes. As a result, parametric models are common in recent works that produce smooth and accurate results [3]. Secondly, smoothness regularizers in variational optical-flow methods are often based on first-order derivatives [6]. These methods are thus biased towards piecewise-linear flow fields limiting their application to fluid flows. This limitation can be addressed by using a second-order regularizer based on the flow fields' divergence and rotation [2,3,7]. However, it is unclear how higher-order regularizers can be designed. In this paper, we propose a parametric model that is robust to noise, is able to represent complicated turbulence, and has a regularizer that is not biased to lower-order flow fields.

Parametric models of fluid flows can be classified into two main groups. The first group are based on physics priors of fluid dynamics, and integrate temporal information into the motion estimation process, producing temporarily consistent results [5]. However, flow fields described by these models are restricted by physics laws, and, as observed in [7], these methods rely on rather involved minimization processes. The second group of methods do not make explicit use of fluid dynamics, but estimate fluid motion solely based on the apparent image deformation, and rely on simple smoothness heuristics to regularize the estimation results [2,12,7]. This group of methods is

M. Kamel and A. Campilho (Eds.): ICIAR 2011, Part I, LNCS 6753, pp. 325–334, 2011.

closely related to the classical problem of optical-flow estimation and nonrigid image registration. A recent work by Isambert et al. [7] produced superior results on turbulent flows, using locally supported vector splines, and representing flows using a multi-scale scheme. However, spline-model optimization can be computationally expensive when dense control-point grids are used, and it is sensitive to local minima. On the other hand, the use of sparse control points can oversmooth estimated flow fields. Most importantly, exact minimization of the functional proposed by Isambert et al. [7] leads to thin-plate splines. This means that their model is still biased to certain lower-order flow fields.

To address the above problems, we introduce a simple parametric model, that is robust yet flexible to represent turbulent flow fields, and can be regularized through a convex functional. Our approach belongs to the second group of methods and makes no assumptions about the fluid's physics properties. Similar to [7], we use a locally supported parametric model to represent a flow field. Instead of using splines and interpolating the motion between control points [7], we use a linear model of orthogonal basis flows represented as holomorphic functions, and approximate the global field by blending the local models using partition-of-unity (Section 2). The use of holomorphic models leads to simpler handling of important fluid-flow properties such as divergence and rotation, and allows us to regularize a fluid flow unbiasedly, by penalizing inconsistencies between neighboring local flows instead of their spatial gradients [7,2]. Additionally, the resulting energy functional is convex, and can be minimized through gradient-descent methods (Section 3). We tested our method on motions from both synthetic and real fluid data (Section 4). Finally, we point out the limitations of our holomorphic flow-field model, and directions for future work (Section 5).

2 Higher-Order Model of Flow Field

Parametric models provide a flexible yet compact flow-field representation. In this section, we represent local flow fields using holomorphic complex functions. Local holomorphic models have been previously used to represent singular points in flow fields [8]. Here, we extend these functions to represent both singular and smooth flow regions.

2.1 Local Flow Field Model

We commence by representing a 2-D vector-flow field as a complex-valued function $F(z)$ defined on a finite domain $\Omega \in \mathbb{C}$ [8,11]. This vector-flow field is then approximated by an holomorphic function centered at $z_0 \in \mathbb{C}$, i.e., $f(z) \approx F(z + z_0)$, that can be modeled using a linear combination of complex basis functions (basis flow fields). For example, the Taylor expansion of $f(z)$ about the origin (i.e., $z_0 = 0$) can be written as a linear combination of complex (orthogonal) monomials $\phi_k(z) = z^k$:

$$f(z) = \sum_{k=0}^{N} a_k \phi_k(z) + R_N(z), \tag{1}$$

where $a_k = \frac{f^{(k)}(0)}{k!}$ are the coefficients, and $R_N(z)$ is the residue. Here, $f^{(k)}(0)$ is the k-th derivative of f evaluated at $z_0 = 0$. For simplicity, we assume the basis $\phi_k(z)$

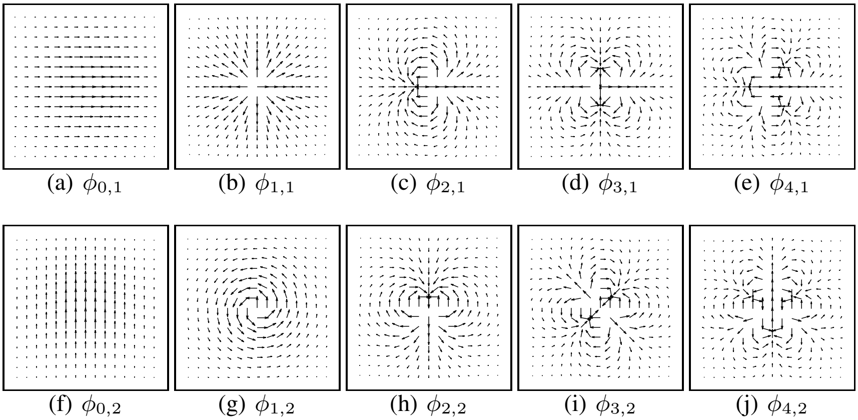


Fig. 1. Basis polynomials $\phi_{k,i}$ multiplied with weight function $w_\sigma(z)$ for $k = 0, \dots, 4$ and $i = 1, 2$. First column: polynomials derived from z^k. Second column: polynomials derived from iz^k. $\phi_{1,1}$ is a rotation-free source field and $\phi_{1,2}$ is a divergence-free vortex. The flow fields exhibit higher-order fluctuation with increasing k.

to be orthogonal, so the coefficients a_k can be calculated by inner product projection. Both the orthogonality condition and projection operator depend on the choice of inner product in the analytic functions space $A(\Omega)$. The classic Hermitian inner product [4] produces complex numbers, making projection calculations difficult. Instead, we use vector fields' correlation [8] as an alternative inner product:

$$\langle f(z), g(z) \rangle = \int_{\mathbb{C}} (f(z) \cdot g(z))\, w_\sigma(z)\, dz, \tag{2}$$

where $\cdot$ is the dot product between two complex numbers, and w_σ is a Gaussian kernel that makes the projection local. Flow-field $f(z)$ can be projected onto the basis function $\phi_k(z)$, with real-domain projection coefficients given by $a_k = \frac{\langle f(z), \phi_k(z) \rangle}{\langle \phi_k(z), \phi_k(z) \rangle}$. Furthermore, we can re-write Equation 2 as:

$$\langle f(z), g(z) \rangle = (F \otimes g)(z_0) = \int_{\mathbb{C}} (F(z + z_0) \cdot g(z))\, w_\sigma(z)\, dz, \tag{3}$$

which can be implemented efficiently using the Fast Fourier Transform (FFT). Given the inner product defined in Equation 2, we can show that complex monomials $\{z^k\}_{k=1}^N$ and $\{iz^k\}_{k=1}^N$ form a complete orthogonal basis. Intuitively, iz^k is a counterclockwise 90-degree rotation of the vectors in z^k. Our basis flows can then be written as: $\phi_{k,1}(z) = z^k$ and $\phi_{k,2}(z) = iz^k$. Figure 1 shows the weighted basis functions $\phi_{k,i} * w_\sigma(z)$ for $k = 0, \dots, 3$. Using (1), the N-th order flow-field approximation at $p \in \Omega$ is:

$$F(z + z_0) \approx f(z) = \sum_{k=0}^{N} \left(a_{k,1}\phi_{k,1}(z) + a_{k,2}\phi_{k,2}(z) \right), \tag{4}$$

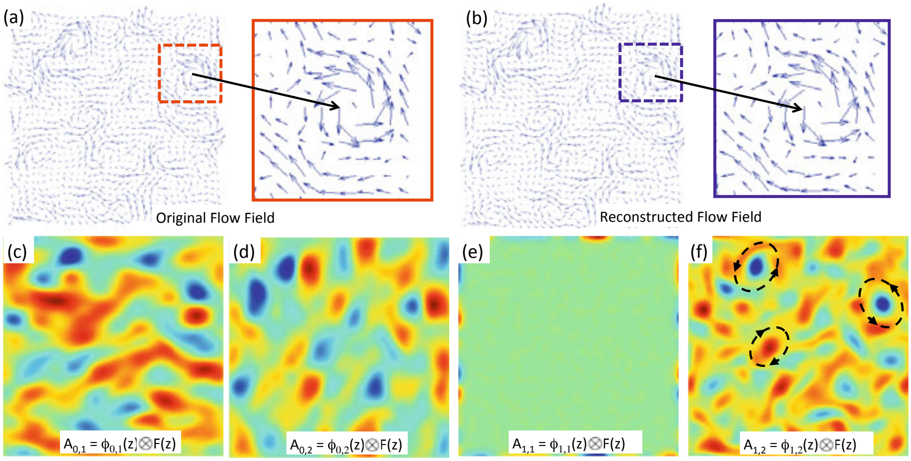


Fig. 2. Decomposition and reconstruction. (a) Original turbulent flow and detail view. (b) Reconstructed flow and detail view. (c)-(f) Correlation coefficient maps for the first four projection coefficients for $\phi_{k,1}(z)$. Coefficient map $A_{1,1}$ in (e) shows that the flow field is divergence free, while stronger responses in $A_{1,2}$ (f) indicate vertex locations. Blue color indicates orientation match between filter and flow data while red indicates reverse orientation.

where $a_{k,i} = \langle f(z), \phi_{k,i}(z) \rangle$, for $k = 1, \ldots, N$, and $i = 1, 2$. The approximation produces $2(N+1)$ real coefficients $a^p = a^p_{0,1}, a^p_{0,2}, \ldots, a^p_{N,1}, a^p_{N,2}$ for location p. According to (3), the coefficients are local values of the cross-correlation between $F(z)$ and $\phi_{k,i}(z)$. It can be shown that by letting $z \longrightarrow 0$ in Equation 4, the local flow field's divergence and rotation simply equal to $a_{1,1}$ and $a_{1,2}$, respectively. This observation shows that both divergence and rotation are represented in our model. Figure 2 shows the correlation between the first two basis pairs and a turbulent flow, i.e., $A_{k,1} = F(z) \otimes \phi_{k,1}(z)$ and $A_{k,2} = F(z) \otimes \phi_{k,2}(z)$, $k = 0, 1$. The turbulent flow field happens to be divergence free so $A_{1,1}$ vanishes almost everywhere. This further confirms that $a_{1,1}$ and $a_{1,2}$ are related to the divergence and rotation of the flow field.

2.2 Blending Local Models into a Global Flow Field

Local flow models can be blended into a global flow field using a partition-of-unity [7]:

$$\widetilde{F}(z) = \sum_{k,i} \int_p A_{k,i}(p) \phi_{k,i}(z-p) h(z-p) dp. \tag{5}$$

Here, function h is a blending function such that $\int h(z)dz = 1$, ensuring that the contributions of neighboring models sum to one (partition-of-unity) [7]. In this paper, we choose $h(z)$ to be a Gaussian function with the same size as our basis flows. This blending approach is more flexible than the interpolating splines [7], as local models are not required to agree at control points. Similarly to splines, the global representation in Equation 5 can be blended using a sparse grid of local models.

3 Fluid Flow Estimation

We now extend the modeling described in previous sections to fluid-flow estimation. In general, fluid-flow estimation is formulated as the following minimization problem [2]:

$$\int D\left(I(\mathbf{x}+\mathbf{v},t+\delta t),I(\mathbf{x},t)\right)d\mathbf{x}+\lambda\int S(\mathbf{v})d\mathbf{x}, \tag{6}$$

where $\mathbf{x}$ and $\mathbf{v}$ are the spatial and velocity vectors, respectively, D is the data term enforcing luminance or mass constancy, and S is the regularizer preferring smooth solutions. Since luminance constancy simplifies computation, and is widely used for incompressible fluid flows, in this work, we enforce the luminance constancy, and leave the mass constancy for future study. The most common data term used to enforce luminance constancy is based on a quadratic form that can be discretized into the well-known optical-flow constraint as $D\left(I(\mathbf{x}+\mathbf{v},t+\delta t),I(\mathbf{x},t)\right) = \left(\nabla I\cdot\mathbf{v}+\partial I/\partial t\right)^2$, where ∇I is the spatial image gradient, and $\frac{\partial I}{\partial t}$ is the time difference. There are two typical regularizers for regularizing the flow fields, including the first-order Horn-Shunk's regularizer [6] and the second-order regularizer used in [2], respectively:

$$S^{(1)}(\mathbf{v})=\|\nabla\mathbf{v}_1\|^2+\|\nabla\mathbf{v}_2\|^2 \text{ and } S^{(2)}(\mathbf{v})=\|\nabla\mathbf{div}(\mathbf{v})\|^2+\|\nabla\mathbf{rot}(\mathbf{v})\|^2. \tag{7}$$

$S^{(1)}$ is widely used in optical flow computation, and is biased towards piecewise linear flows, while $S^{(2)}$ is considered more appropriate for regulating fluid motions. Here, since we represent flow fields using parametric models, instead of recovering $\mathbf{v}$ directly, we aim at finding the optimal coefficients representing the underlying motion between two images. In the following section, we first show how the optical-flow constraint and the existing regularizers can be rewritten using the proposed model. Then, we introduce a general regularizer for arbitrary-order flow fields.

3.1 Local Optical-Flow Constraint

Let us write the image gradient as a complex function $\nabla I(z)=\frac{\partial I}{\partial x}+\frac{\partial I}{\partial y}i$, and let $f(z)$ represent $\mathbf{v}$ at pixel p. We can then substitute the linear approximation of $f(z)$ in (4) into the optical-flow constraint to minimize the following weighted error function:

$$D(p)=\sum_{z\in N_p}w_\sigma(z)\left(\sum_{k=0}^{N}\left(a_{k,1}\phi_{k,1}(z)+a_{k,2}\phi_{k,2}(z)\right)\cdot\nabla I(z)-\partial I/\partial t\right)^2, \tag{8}$$

where $w_\sigma(z)$ is a Gaussian function that weights the image evidence more at the center. Equation 8 can be written in a compact matrix form: $(\mathbf{R}_p\mathbf{a}_p-\mathbf{T}_p)^{\mathsf{T}}\mathbf{W}(\mathbf{R}_p\mathbf{a}_p-\mathbf{T}_p)$, with $\mathbf{W}$ contains the weighting factor $w_\sigma(z)$, $\mathbf{R}_p$ is calculated from $\phi_{k,i}\cdot\nabla I(z)$, and $\mathbf{T}_p$ is obtained by stacking $\frac{\partial I}{\partial t}$. Minimizing Equation 8 leads to a local optical flow calculation similar to the Lucas-Kanade [9] method that can be solved as a linear system.

3.2 Global Smoothness Constraint

In this section, we show how smoothness constraints can be formulated directly from the local coefficients vector $\frac{\partial a^p}{\partial p}, p\in\Omega$. If we consider $z=0$ in Equation 4, the local velocity vector is simply $(a_{0,1}, a_{0,2})$. Thus, the first-order regularizer can be written as $S^{(1)}(\mathbf{v}) = \|\nabla a_{0,1}\|^2 + \|\nabla a_{0,2}\|^2$. Furthermore, we haven shown in Section 2.1 that $\mathbf{div} f(z) = a_{1,1}$ and $\mathbf{rot} f(z) = a_{1,2}$ when $z \to 0$. As a result, the second-order regularizer becomes: $S^{(2)}(\mathbf{v}) = \|\nabla a_{1,1}\|^2 + \|\nabla a_{1,2}\|^2$. Similarly, we can define an arbitrary-order regularizer $S_1^N = \sum_{k=0}^{N} \beta_k \left(\|\nabla a_{k,1}\|^2 + \|\nabla a_{k,2}\|^2\right)$ where $\beta_k \geq 0$ are weight factors that emphasize on different orders, or equivalently:

$$S_1^{(n)} = \sum_{z \in N_p} \left(\mathbf{a}_z - \mathbf{a}_p\right)^{\mathsf{T}} \mathbf{\Gamma} \left(\mathbf{a}_z - \mathbf{a}_p\right), \tag{9}$$

where $\mathbf{\Gamma} = \mathrm{diag}(\beta_0, \ldots, \beta_N)$, and N_p is the set of neighboring local models. By choosing small β_k for lower-order coefficients, we avoid penalizing lower-order variations. This can be justified by noticing that $a_{k,i}$ is related to the flow field's derivatives through Taylor's expansion in (1), and the n-th order spatial derivatives of $F(z)$ can be measured from the derivatives of the corresponding coefficients $a_{n,i}, i = 1, 2$.

Unbiased higher-order regularizer. The regularizer in (9) penalizes spatial variations of model parameters, and is similar to the one used in [10]. However, penalizing model parameters' gradients may lead to bias towards certain orders of flow fields, depending on the choice of the weighting parameters $\mathbf{\Gamma}$. Also, simply penalizing model parameters' spatial variations ignores the fact that the variation can be partially caused by local coordinate system shifting. For example, the local flow $f(z) = z + z^2$ observed at a neighboring position $z + \delta z$ will be $f(z + \delta z) = (\delta z)^2 + (1 + 2\delta z)z + z^2$. In other words, there will be model parameter variations even when the flow field follows exactly a polynomial model. We account for these variations by shifting the local parameters before comparison with neighboring models. Fortunately, we can write the shifting of a basis function (monomial) z^k as a linear combination of lower-order monomials, i.e., $(z+\delta z)^k = (\delta z)^k + (\delta z)^{k-1} z + \ldots + z^k$. As result, the shifting operator can be written as a lower-triangular matrix $H(\delta z)$. Thus, an alternative regularizer can be defined as:

$$S_2^{(n)} = \sum_{z \in N_p} \left(\mathbf{H}(p-z)\mathbf{a}_z - \mathbf{a}_p\right)^{\mathsf{T}} \mathbf{\Gamma} \left(\mathbf{H}(p-z)\mathbf{a}_z - \mathbf{a}_p\right), \tag{10}$$

where $\mathbf{H}(p - z)$ is the shifting matrix, and the weighting matrix $\mathbf{\Gamma}$ is used to make the notation consistent with Equation 9. In this paper, we simply choose $\mathbf{\Gamma} = \lambda \mathbf{I}$ with $\lambda > 0$, for both (9) and (10). In this way, we are not penalizing the spatial variations of the model parameters. Instead, we penalize the *inconsistency* between local models, so flow fields with different orders will not be biased by the regularizer for the *magnitude* of their variations, as long as they make *consistent* variations. It is easy to verify that any holomorphic functions (flows) with order less than N can make $S_2^{(n)}$ vanish, and this confirms that lower- and higher-order flow fields are equally penalized.

3.3 Gradient-Descent Minimization

We now combine both local and global constraints into a single functional as:

$$E_1^{(n)} = \sum_{p \in \Omega} D(p) + S_1^{(n)}(p) \quad \text{or} \quad E_2^{(n)} = \sum_{p \in \Omega} D(p) + S_2^{(n)}(p). \tag{11}$$

Here, both $E_1^{(n)}$ and $E_2^{(n)}$ are convex, and can be minimized using variational calculus. Since their minimizing procedures are analogous, we will only explain the minimization for $E_2^{(n)}$. The gradients for this functional can be derived as follows:

$$\frac{\partial E^N}{\partial \mathbf{a}_p} = 2\left\{ \mathbf{a}_p^{\mathsf{T}} \underbrace{(\mathbf{R}_p^{\mathsf{T}}\mathbf{W}\mathbf{R}_p + \|N_p\|\mathbf{\Gamma})}_{\mathbf{M}_p} - \sum_{z\in N_p} \mathbf{a}_z^{\mathsf{T}} \underbrace{\mathbf{H}^{\mathsf{T}}(p-z)\mathbf{\Gamma}}_{\text{shifting term}} - \underbrace{\mathbf{T}_p^{\mathsf{T}}\mathbf{W}\mathbf{R}_p}_{\mathbf{N}_p} \right\}. \quad (12)$$

In (12), matrices $\mathbf{M}_p$ and $\mathbf{N}_p$ is pre-calculated from image gradients and basis flows. The same applies to the shifting term $\mathbf{H}^{\mathsf{T}}(p-z)\mathbf{\Gamma}$.

4 Experiments

The goal of our experiments is to show that fluid-motion estimation can be improved using our high-order model. We began by evaluating the homomorphic model by obtaining decompositions and reconstructions on synthetic turbulent flows. Then, we ran our fluid-motion estimation algorithms on both synthetic and real images. In all implementations, we used luminance-constancy instead of mass-constancy constraint. The reconstruction's average end-point error (APE) on European FLUID dataset [1] using 2nd-order and 3dr-order models as a function of basis-flow radius was less than 5%, showing that the fluid motion was well represented by our model. It is worth noticing that as the radius of the local models approaches zero, our representation becomes over-parameterized, and the 3rd-order model produced larger reconstruction error for the radius were smaller than two pixels.

Synthetic PIV images. On synthetic images, we quantitatively compared the following methods: the classic Horn-Shunk method [1] ($S^{(1)}$), a B-spline adaptation of the method in [7], and also with our higher-order regularizer without shifting ($E_1^{(n)}$) and with shifting ($E_2^{(n)}$). For the last two, we tested the cases of $n = 2$ and $n = 3$. Although we do not have implementations of the second-order regularizer used in [2] (i.e., $S^{(2)}$), our regularizer $E_1^{(2)}$ can be seen as a parametric version of it. As ground truth is hard to obtain for fluid images, we resorted to synthetic PIV images from the FET-Open European project FLUID [1]. This database contains 6 different types of stable flows, and turbulent flows. As stable and turbulent flows are different in nature, we tuned the algorithm parameters separately for each dataset. These parameters are: (1) the smoothness weight λ_{hs} for Horn-Shunk's method; (2) the spacing of control points d_{sp}, and the smoothness weight λ_{sp} for spline-based method; (3) for our method, the scale (radius) of the parameterized model r, the spacing between local models d, and the regularizer weight λ. Table 1 summarizes the parameters used for each method and dataset.

Tables 2 show the average angular error (AAE) and average end-point error (APE) of the compared methods. Our method performed better on almost all sequences, and the errors decrease with increasing approximation order. Comparing results of $E_1^{(n)}$ and $E_2^{(n)}$ shows that the shifting operator increases estimation accuracy. Additionally, the

[1] Available for download from: `http://www.cs.brown.edu/~dqsun/`

Table 1. Algorithm Parameters

Dataset	Horn-Shunk	Spline		$E_1^{(2)}\&E_2^{(2)}$			$E_1^{(3)}\&E_2^{(3)}$		
	λ_{hs}	λ_{sp}	d_{sp}	λ	d	r	λ	d	r
Stable Flows	2500	0.1	32	0.1	8	32	0.5	8	32
Turbulence	1500	0.1	8	0.1	2	6	0.5	2	6

Table 2. AAE and APE on Analytic Fluid Sequence

	Seq. 1		Seq. 2		Seq. 3		Seq. 4		Seq. 5		Seq. 6		Turb.	
	AAE	APE	AAE	APE	AAE	APE	AAE	APE	AAE	APE	AAE	APE	AAE	APE
HS	1.02	0.04	1.96	0.04	1.01	0.04	2.75	0.06	2.77	0.06	1.62	0.05	22.09	0.43
Spline	0.63	0.03	0.96	0.02	1.13	0.04	2.73	0.06	2.28	0.05	1.43	0.05	7.27	0.13
$E_1^{(2)}$	0.85	0.04	1.70	0.04	0.78	0.03	2.48	0.05	2.59	0.05	1.34	0.04	4.62	0.09
$E_2^{(2)}$	0.80	0.03	1.63	0.03	0.72	0.03	2.43	0.05	2.53	0.05	1.31	0.04	4.62	0.08
$E_1^{(3)}$	0.58	0.03	1.38	0.03	0.63	0.03	1.87	0.04	1.88	0.04	1.45	0.06	4.58	0.08
$E_2^{(3)}$	0.58	0.03	1.24	0.02	0.55	0.02	1.87	0.04	1.89	0.04	1.49	0.18	4.30	0.08

experiments confirmed the observation in [7] that spline-based methods produce better results than their nonparametric counterparts, especially on turbulent flows. Figure 4 shows streamline and vorticity maps of the extracted turbulence motion. Although the difference is visually small from the streamlines, it can be seen that both the spline-based and our method produce a 'smoother' vorticity map than the Horn-Shunk method, and that our method's vorticity map is closest to the ground truth in its magnitude.

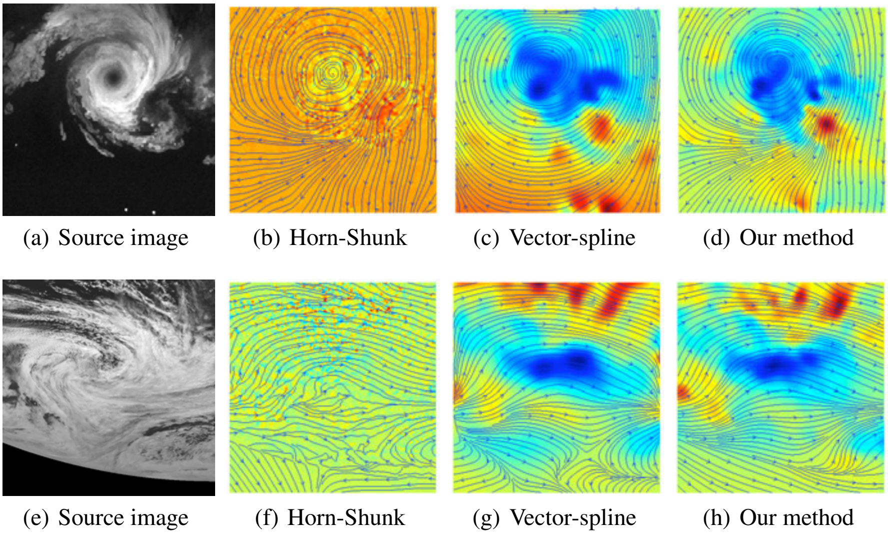
Real-world images. In Figure 4, we show the estimated motion from a wingtip vortex [2] and satellite images [3]. Both the spline model and ours produce smoother results than the Horn-Shunk method. However the spline model easily got trapped in local minima when small smoothness parameters were used, and produced over-smoothed results when the parameter was large. Specifically, for satellite images, all the three methods produced a weak flow for static image regions due to the smoothness constraint. Interestingly, as we have discussed in Section 3.2, Horn-Shunk's first-order regularizer produced piecewise linear flows, and the spline model split the flow field to satisfy the thin-plate deformation energy, while ours produced consistent background flow.

5 Limitations of Our Method and Future Work

We have proposed a higher-order model of flow fields using complex polynomials. Using this model, we were able to reformulate the optical flow computation in a general

[2] Courtesy of ONERA

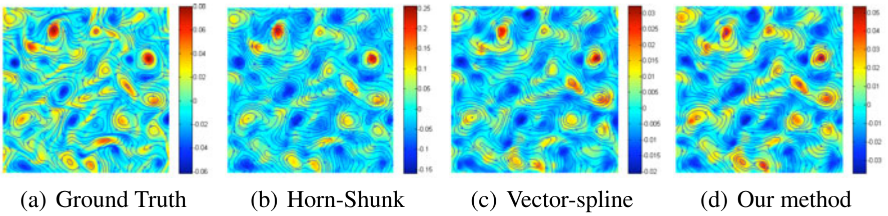
[3] Copyright @ EUMETSAT



(a) Source image (b) Horn-Shunk (c) Vector-spline (d) Our method

(e) Source image (f) Horn-Shunk (g) Vector-spline (h) Our method

Fig. 3. Real-world image sequences. The first row shows flow fields estimated from a Wingtip Vortex, and the second row shows the ones from satellite images. Compared to the spline model, our method does not over-smooth the flow fields, and produce more consistent results.



(a) Ground Truth (b) Horn-Shunk (c) Vector-spline (d) Our method

Fig. 4. Fluid motion estimation. Both spline-based methods and ours produced smoother results than Horn-Shunk's. The vorticity estimated by our method is closer to the ground truth.

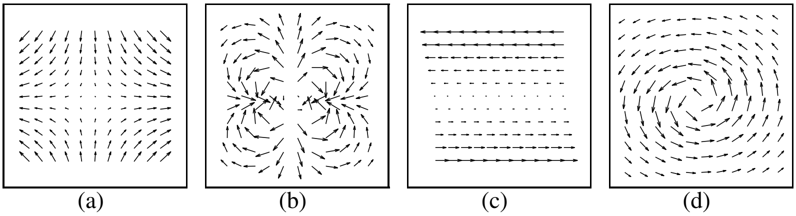


Fig. 5. Flow fields that cannot be well approximated by holomorphic functions of similar scales, including a conjugate flow $f(z) = \overline{z}$ (a) and its holomorphic approximation shown in (b), a shear flow $f(z) = z + \overline{z}$ (c) and its holomorphic approximation shown in (d)

way in which the regularizer can be chosen to penalize certain orders of variations. It is important to point out that the holomorphic assumption used in our approximation model is restrictive as certain flow fields may not be well represented by our model.

Figure 5 shows two examples of such flows, namely, the conjugate flow, $f(z) = \overline{z}$, and the affine flow, $f(z) = z + \overline{z}$, with their holomorphic approximations using basis flows of similar scales to the approximated local flows. Both of the flows are non-analytic anywhere in the complex plane, and their holomorphic approximations are poor. This problem can be partially addressed by minimizing the basis flows' scales. In the extreme case when the bases' scale approaches zero, our flow-field model become over-parameterized, and the flow fields can be fully represented. However, this would increase computational cost, and we believe the better solution lies in extending our approximation model to include non-analytic basis flows. Our future work also includes extension of the method to 3-D flow-field estimation, integration with flow-field singular pattern detection [8], and the usage of mass-constancy constraints [2].

References

1. Carlier, J.: Second set of fluid mechanics image sequences. European Project 'Fluid image analysis and description', FLUID (2005), http://www.fluid.irisa.fr/
2. Corpetti, T., Memin, E., Prez, P.: Dense estimation of fluid flows. IEEE Trans. Patt. Anal. and Mach. Intel. 24(3), 365–380 (2002)
3. Cuzol, A., Hellier, P., Memin, E.: A low dimensional fluid motion estimator. International Journal of Computer Vision 75(3), 329–349 (2007)
4. Davies, B.: Integral Transforms and Their Applications. Springer, Heidelberg (2002)
5. Heitz, D., Memin, E., Schnorr, C.: Variational fluid flow measurements from image sequences: synopsis and perspectives. Experiments in Fluids 48(3), 369–393 (2009)
6. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17(1-3), 185–203 (1981)
7. Isambert, T., Berroir, J.-P., Herlin, I.: A multi-scale vector spline method for estimating the fluids motion on satellite images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 665–676. Springer, Heidelberg (2008)
8. Liu, W., Ribeiro, E.: Scale and Rotation Invariant Detection of Singular Patterns in Vector Flow Fields. In: S-SSPR, pp. 522–531 (2010)
9. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI, pp. 674–679 (1981)
10. Nir, T., Bruckstein, A.M., Kimmel, R.: Over-parameterized variational optical flow. Int. J. Comput. Vision 76(2), 205–216 (2008)
11. Petrila, T., Trif, D.: Basics of fluid mechanics and introduction to computational fluid dynamics. Springer, Heidelberg (2005)
12. Suter, D.: Motion estimation and vector splines. In: CVPR, pp. 939–942 (2002)