Liu, T 2017 OpenOpticalFlow: An Open Source Program for Extraction of Velocity Fields from Flow Visualization Images. *Journal of Open Research Software,* 5: 29, DOI: https://doi.org/10.5334/jors.168

SOFTWARE METAPAPER

OpenOpticalFlow: An Open Source Program for Extraction of Velocity Fields from Flow Visualization Images

Tianshu Liu
Department of Mechanical and Aerospace Engineering, Western Michigan University, Kalamazoo, MI 49008, US
tianshu.liu@wmich.edu

This paper describes OpenOpticalFlow, an open source optical flow program in Matlab for extraction of high-resolution velocity fields from various flow visualization images. This program is a useful tool for researchers to use the optical flow method in various flow measurements. The principles of the optical flow method are concisely described, including the physics-based optical flow equation, the variational solution, and errors. The central part of this paper is the descriptions of the main program, relevant subroutines and selection of the relevant parameters in optical flow computation. Examples are given to demonstrate the applications of the optical flow method.

Keywords: optical flow; velocimetry; variational solution; flow visualization; fluid mechanics; measurement technique; image processing; Matlab

(1) Overview

Introduction

The optical flow method has been developed for extraction of high-resolution velocity fields from various images of continuous patterns from flow visualization images obtained in laboratories to cloud and ocean images taken by satellites/spacecraft [1–6]. The rational foundation for application of the optical flow method to fluid flow measurements is the quantitative connection between the optical flow and the fluid flow velocity for various flow visualizations. Liu and Shen [7] have derived the projected motion equations for various flow visualizations including laser-sheet-induced fluorescence images, transmittance images of passive scalar transport, schlieren, shadowgraph and transmittance images of density-varying flows, transmittance and scattering images of particulate flows, and laser-sheet-illuminated particle images. Further, these equations are recast into the physics-based optical flow equation in the image plane. Physically, the optical flow is proportional to the light-ray-path-averaged velocity of fluid (or particles) weighted in a relevant field quantity like dye concentration, fluid density or particle concentration. The optical flow method has been used to study the flow structures of Jupiter's Great Red Spot (GRS), impinging jets, and laser-induced underwater shock wave [8–11]. A mathematical analysis of the variational solution of the optical flow and an iterative numerical algorithm are given by Wang et al. [12]. The systematical error analysis of the optical flow method in velocity measurements is given by Liu et al. [11] in comparison with the well-established cross-correlation method in particle image velocimetry (PIV).

The objective of this paper is to describe an open source optical flow program in Matlab for extraction of high-resolution velocity fields from various flow visualization images. It would help researchers to apply the optical flow method to their specific problems in relevant scientific and engineering fields. First, the principles of the optical flow method are briefly described, including the physics-based optical flow equation, the variational method, and a short error analysis. The physical meaning of the optical flow in fluid-mechanic measurements is interpreted. Then, the main program and the key subroutines are described, particularly on the selection of the relevant parameters in optical flow computation. Next, to demonstrate the applications of this program, several examples based on simulated and real flow visualization images are presented.

Principles of optical flow method

a. Physics-based optical flow equation

Flow visualization techniques are widely used based on the applications of observable tracers (such as particles and dyes) and change of certain physical properties of fluid (such as the density). Digital flow visualization images are usually obtained by using cameras detecting radiation with a certain wavelength bandwidth from a fluid medium in a flow. Liu and Shen [7] have given the physics-based optical flow equation for different flow visualizations, which is written in in the image coordinates as:

$$\frac{\partial g}{\partial t} + \nabla \cdot \left(g\boldsymbol{u} \right) = f(x_1, x_2, g), \tag{1}$$

where $\boldsymbol{u} = (u_1, u_2)$ is the velocity in the image plane referred to as the optical flow, g is the normalized image intensity that is proportional to the radiance received by a camera, $\nabla = \partial/\partial x_\beta$ is the spatial gradient, $\nabla^2 = \partial^2/\partial x_\beta \partial x_\beta$ is the Laplace operator, and $f(x_1, x_2, g)$ is related to the boundary term and diffusion term. The optical flow $\boldsymbol{u} = (u_1, u_2)$ is proportional to the light-path-averaged velocity weighted with the field quantity ψ related to a visualizing medium [7]. In the special case where $g\nabla\cdot\boldsymbol{u} = 0$ and $f = 0$, Eq. (1) is reduced to the Horn-Schunck brightness constraint equation $\partial g/\partial t + \boldsymbol{u}\cdot\nabla g = \boldsymbol{0}$ [13]. In general, the optical flow is not divergence-free, i.e., $\nabla\cdot\boldsymbol{u} \neq 0$.

b. Variational solution

To determine the optical flow, a variational formulation with a smoothness constraint is typically used [7, 13]. Given g and f, we define a functional:

$$\begin{aligned} J(\boldsymbol{u}) = &\int_\Omega \left[\partial g / \partial t + \nabla \cdot \left(g\boldsymbol{u} \right) - f \right]^2 dx_1 dx_2 \\ &+ \lambda \int_\Omega \left(\left| \nabla u_1 \right|^2 + \left| \nabla u_2 \right|^2 \right) dx_1 dx_2, \end{aligned} \tag{2}$$

where λ is the Lagrange multiplier and Ω is an image domain. By minimizing the functional, i.e., $J(\boldsymbol{u}) \to min$, we obtain the Euler-Lagrange equation:

$$g\nabla \left[\partial g / \partial t + \nabla \cdot (g\boldsymbol{u}) - f \right] + \lambda \nabla^2 \boldsymbol{u} = 0. \tag{3}$$

The standard finite difference method is used to solve Eq. (3) with the Neumann condition $\partial\boldsymbol{u}/\partial n = 0$ on the image domain boundary $\partial\Omega$ for the optical flow [7, 12].

c. Errors

An error analysis of the optical flow method was given by Liu et al. [11]. The general estimate for the total error of optical flow computation is given by:

$$\varepsilon = \left\| \Delta \boldsymbol{x} \right\| \sqrt{\frac{c_1}{\left\| \nabla g \right\|^2 \left\| \boldsymbol{u} \right\|^2} + c_2 \left\| \nabla \boldsymbol{u} \right\|^2}, \tag{4}$$

Where the relevant parameters are the displacement magnitude $\|\Delta\boldsymbol{x}\|$, the image intensity gradient magnitude $\|\nabla g\|$, the velocity gradient magnitude $\|\nabla\boldsymbol{u}\|$ and the velocity magnitude $\|\boldsymbol{u}\|$, and c_1 and c_2 are coefficients to be determined. Since according to Eq. (4) the total error is proportional to $\|\Delta\boldsymbol{x}\|$, small displacements are generally required for a good accuracy in optical flow computation. A rule of thumb is that $\|\Delta\boldsymbol{x}_p\|$ should be smaller than the characteristic size of patterns (e.g. particles in PIV). The error would be larger in the regions where $\|\nabla\boldsymbol{u}\|$ is large and/or $\|\nabla g\|$ is smaller.

For particle images in PIV, Eq. (4) can be further evaluated by including the effects of the particle image diameter and the particle image density. Therefore, the estimate for the total error of the optical flow method applied to PIV images is given by:

$$\varepsilon = \left\| \Delta \boldsymbol{x}_p \right\| \sqrt{\frac{c_1 d_p^2}{\left\| \boldsymbol{u}_p \right\|^2} + c_2 \left\| \nabla \boldsymbol{u}_p \right\|^2 + \frac{c_3}{d_p^2} + \frac{c_4}{N_p^m} \frac{\left\| \Delta^m \boldsymbol{x}_p \right\|^2}{\left\| \Delta \boldsymbol{x}_p \right\|^2}}, \tag{5}$$

where c_1, c_2, c_3 and c_4 are coefficients to be determined, and Δ^m is the mth-order difference operator. In Eq. (5), the main parameters are the particle displacement magnitude $\|\Delta\boldsymbol{x}_p\|$, the particle image diameter d_p, the particle velocity gradient magnitude $\|\nabla\boldsymbol{u}_p\|$, and the particle image density N_p. In general, the particle displacement magnitude $\|\Delta\boldsymbol{x}_p\|$ should be small in optical flow computation for PIV images, particularly in regions of large velocity gradients. The particle image diameter d_p could have the optimal value for optical flow computation. The particle image density N_p should be suitably large, and there would be the optimal value of N_p.

Implementation and architecture

a. General description

The optical flow program is written in Matlab, and the files are contained in a folder named "OpenOpticalFlow.v1". The main program is "Flow_Diagnostics_Run.m". In the main program, a pair of images is loaded, the relevant parameters are set, images are pre-processed, optical flow computation with a coarse-to-fine iteration is conducted, and the results are plotted. The subroutine for solving Eq. (3) is "liu_shen_estimator.m". In computations, the solution of the Horn-Schunck optical flow equation ("horn_schunck_estimator.m") is used as an initial approximation for Eq. (3) for faster convergence. The combination of "liu_shen_estimator.m" and "horn_schunck_estimator.m" constitutes the key elements of the optical flow program.

The optical flow processing can be made in a selected rectangular domain of interest for regional flow diagnostics when the index "index_region" is set at 1. Otherwise, the processing is conducted in the whole image domain when the index "index_region" is set at 0. The input files and relevant parameters for optical flow computation are listed in **Table 1**. The image files "Im1" and "Im2" could be in the conventional image format such as "tif", "bmp", or "jpeg". The input parameters are discussed in the following subsections. The optical flow $\boldsymbol{u} = (u_1, u_2)$ is calculated by solving Eq. (3). **Table 2** lists the outputs. The output data are "ux0" and "uy0" that are the coarse-grained velocity components in the image plane (x, y), which are extracted from the dowmsampled images in the coarse-to-fine process. The refined velocity components "ux_corr" and "uy_corr" are obtained by using the coarse-to-fine scheme. The unit of the velocity given by the program is pixels/unit-time, where the unit time is the time interval between two input images. The velocity can be converted to the physical unit of m/s after the relation between the image plane and the 3D object space is established through camera calibration/orientation. The vorticity, strain rate and second invariant can be calculated based

Table 1: Inputs.

Input files and parameters	Notation	Unit	Note
image pair	Im1, Im2	–	tif, bmp, jpeg etc.
Lagrange multipliers	lambda_1 for the Horn-Schunck estimator lambda_2 for the Liu-Shen estimator	–	regularization parameter in variational solution
Gaussian filter size	size_filter	pixel	removing random image noise
Gaussian filter size	size_average	pixel	correction for local illumination intensity change
scale factor for downsampling of original images	scale_im	–	reduction of initial image size in coarse-to-fine scheme
number of iterations	no_iteration	–	iteration in coarse-to-fine scheme
indicator for regional diagnostics (0 or 1)	index_region	–	"0" for whole image; "1" for selected region

Table 2: Outputs.

Output files	Notation	Unit	Note
coarse-grained velocity	ux0, uy0	pixels/unit-time	based on downsampled images
refined velocity	ux_corr, uy_corr	pixels/unit-time	refined result with full spatial resolution in coarse-to-fine process

on a high-resolution velocity field. Furthermore, this program can be adapted for processing of a sequence of image pairs such that the statistical quantities of the flow can be obtained.

b. The Lagrange multipliers

The optical flow program uses the Horn-Schunck estimator ("horn_schunck_estimator.m") for an initial solution and Liu-Shen estimator ("liu_shen_estimator.m") for a refined solution of Eq. (1). In the main program, the Lagrange multipliers "lambda_1" and "lambda_2" are selected for the Horn-Schunck and Liu-Shen estimators, respectively. For example, "lambda_1" = 20 and "lambda_2" = 2000 for typical computations.

There is no rigorous theory for determining the Lagrange multiplier in the variational formulation of the optical flow equation. The Lagrange multiplier acts like a diffusion coefficient in the corresponding Euler-Lagrange equations. Therefore, a larger Lagrange multiplier tends to smooth out finer flow structures. In general, the smallest Lagrange multiplier that still leads to a well-posed solution could be selected by a trial-and-error process. However, within a considerable range of the Lagrange multipliers, the solution is not significantly sensitive to its selection. Simulations based on a synthetic velocity field for a specific measurement could be carried out to determine the Lagrange multiplier by using an optimization scheme [8].

c. Filtering

Pre-processing of images is sometimes required to remove the random noise by using a Gaussian filter. The standard deviation (std) of a Gaussian filter is selected, depending on the noise level in a specific application (for example the std of a Gaussian filter is 4–6 pixels for images of 480 × 520 pixels). In the main program, the mask size of a Gaussian filter is given by the parameter called "size_filter", and the std of the Gaussian filter is 0.6 of the mask size.

d. Correction for illumination change

The underlying assumption in optical flow computation is that the illumination light intensity keeps constant (or time-independent) in flow visualization, which is valid in the well-controlled laboratory conditions. In some situations, however, the illumination intensity field could change in a time interval between two successively acquired images. For example, when the Jupiter's atmosphere structures were imaged by spacecraft, the illumination intensity field from the Sun could change in a relatively long time interval (hours) during image acquisition depending on the relative movement between the Sun, Jupiter and spacecraft. In this case, correction for this illumination intensity change is required before applying the optical flow method to these images.

In the first step in this subroutine, the overall illumination change in the whole image is corrected by normalizing both the images. In the second step, a simple scheme for correcting local illumination intensity changes is also based on the application of a Gaussian filter [11]. The selection of the std (or size) of a Gaussian filter is important to determine the local averaged intensity value for correction. This size of the Gaussian filter is given by the parameter called "size_average" in the main program.

When the size of a filter is too large, the local variation associated with the illumination intensity change in images cannot be corrected. On the other hand, when the filter size is too small, the apparent motion of patterns/features in the two images would be artificially reduced after the procedure is applied. The selection of the filter size is a trial-and-error process depending on the pattern of illumination change in a specific measurement, and simulations on a synthetic velocity field are used to determine the suitable size.

e. Coarse-to-fine scheme

As indicated in Section 2.3, the optical flow method as a differential approach is more suitable for extraction of high-resolution velocity fields with small displacement vectors from images of continuous patterns (typically less than 5 pixels depending on the size of image patterns). When the displacements in the image plane are large (for example more than 10 pixels), the error in optical flow computation would be significant. In this case, a coarse-to-fine iterative scheme can be used, which is implemented in a loop in the main program.

First, the original images are suitably downsampled by a suitable scale factor (such as 0.5) so that the displacements in pixels are small enough (1–5 pixels). A scale factor is specified by the parameter called "scale_im" for downsampling. For example, "scale_im" = 0.5 means that the original images are reduced to 50% of the original size. A coarse-grained velocity field is obtained by applying the optical flow algorithm to the downsampled images.

Then, the resulting coarse-grained velocity field is used to generate a synthetic shifted image with the same spatial resolution as the original image #1 (i.e., the first one in the two successive images) by using an image-shifting (or image-warping) algorithm in which a translation transformation is used for large displacements and the discretized optical flow equation is used for sub-pixel correction.

Next, the velocity difference field between the synthetically shifted image and the original image #2 is determined by using the optical flow algorithm, and it is added on the initial velocity field for correction or improvement. Thus, a refined velocity field is successively recovered by iterations to achieve a better accuracy. The iteration number is given by "no_iteration". Usually, one or two iteration is sufficient. When "no_iteration" is set at 0, the coarse-to-fine scheme is disabled.

Examples

a. Oseen vortex pair in uniform flow

To demonstrate the application of this program, simulations are conducted on particle images (PIV images) in a synthetic flow 3:4 an Oseen vortex pair in uniform flow. A sample particle image with the size of 500 × 500 pixels and the 8-bit dynamic range is generated, where 10000 particles with a Gaussian intensity distribution with the standard deviation of $\sigma = 2$ pixels are uniformly distributed. **Figure 1** shows a pair of the generated particle image that is used for simulations, where the mean characteristic image diameter of particles is $d_p = 4$ pixels and the particle image density is 0.04 1/pixel^2. A synthetic velocity field, generated by superposing an Oseen vortex pair on a uniform flow, is used as a canonical flow. Two Oseen vortices are placed at $(m/3,\ n/2)$ and $(2m/3,\ n/2)$ in an image, respectively, where m (500) and n (500) are the numbers of rows and columns of the image. The circumferential velocity of an Oseen vortex is given by $u_\theta = (\Gamma / 2\pi r)[1 - exp(-r^2 / r_0^2)]$, where the vortex strengths are $\Gamma = \pm 7000$ (pixel)^2/s and the vortex core radius is $r_0 = 15$ pixels. The uniform flow velocity is 10 pixels/s. The second image is generated by deforming the original synthetic particle image based on the given velocity field after a time step Δt. This processing is made by applying an image-shifting algorithm, which faithfully describes the motion of image patterns for a given velocity field since the physics-based optical flow equation is derived from the governing transport equations for flow visualizations. The time step Δt is used to control the maximum displacement in synthetic particle images. This flow with particle images has been used as a typical case in the simulations to evaluate the accuracy of the optical flow method [11].

In optical flow computation, the Lagrange multiplier in the Horn-Schunck estimator is set at 20, which provides a good initial solution for refinement by the Liu-Shen estimator based on Eq. (1). The Lagrange multiplier in the Liu-Shen estimator is fixed at 2000 for a refined velocity

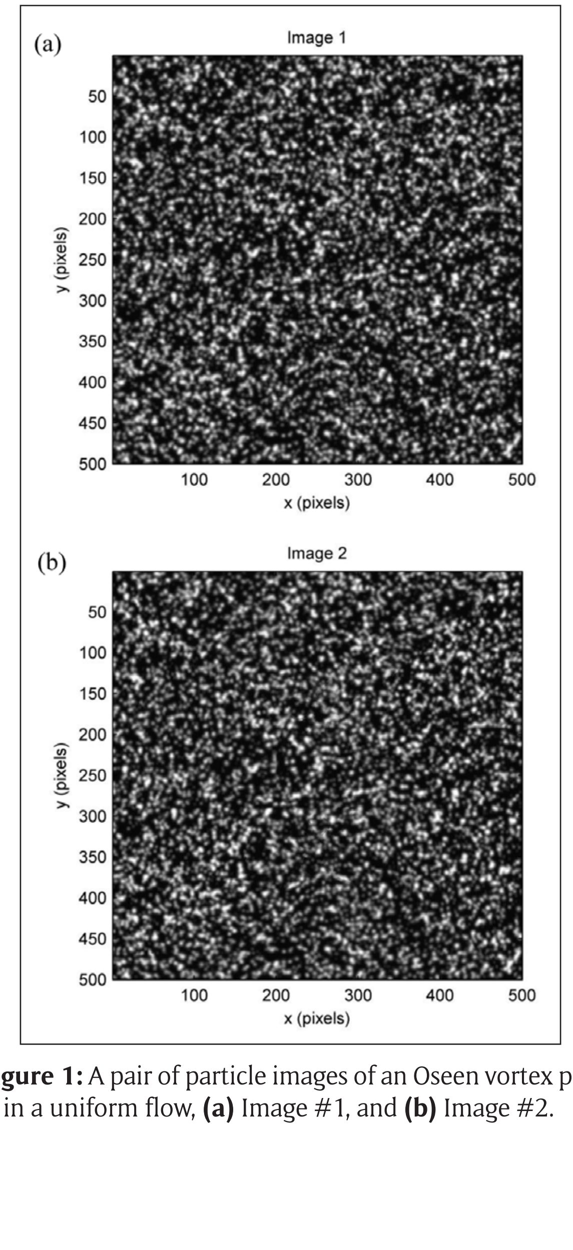

Figure 1: A pair of particle images of an Oseen vortex pair in a uniform flow, **(a)** Image #1, and **(b)** Image #2.

field, and it does not significantly affect the velocity profile in a range of 1000–20000 except the peak velocity near the vortex cores in this flow. A pair of particle images with 10000 particles in **Figure 1** is processed as a typical case, where the maximum displacement is 2.6 pixels. **Figure 2a** and **b** show the coarse-grained and refined velocity vector fields extracted from the particle images, respectively. The optical flow method gives 500 × 500 vectors from the original images. **Figure 3** shows the corresponding streamlines. **Figure 4** shows the refined vorticity field normalized by its maximum value.

The accuracy of the optical flow method applied to PIV images has been studied through simulations and experiments in comparison with the well-established cross-correlation method [11]. As indicated before, the uncertainty in optical flow computation depends on the four parameters: particle displacement, particle velocity gradient, particle image density, and particle image diameter. An Oseen vortex pair in uniform flow has been used as a typical case to quantitatively evaluate the optical flow errors in the parametric space through

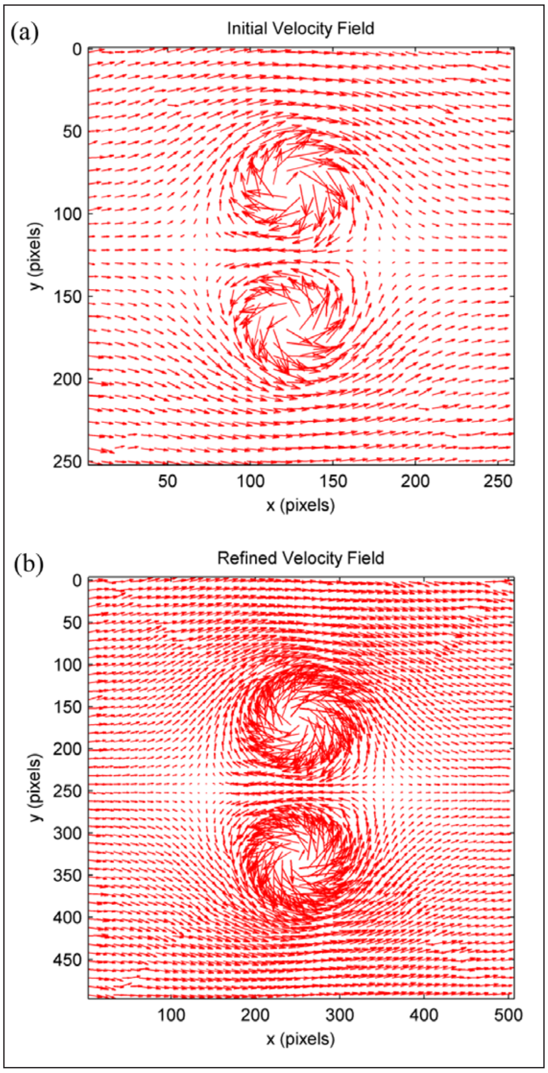

Figure 2: Velocity vectors of an Oseen vortex pair in a uniform flow extracted from the particle images, **(a)** Coarse-grained field, and **(b)** Refined field.

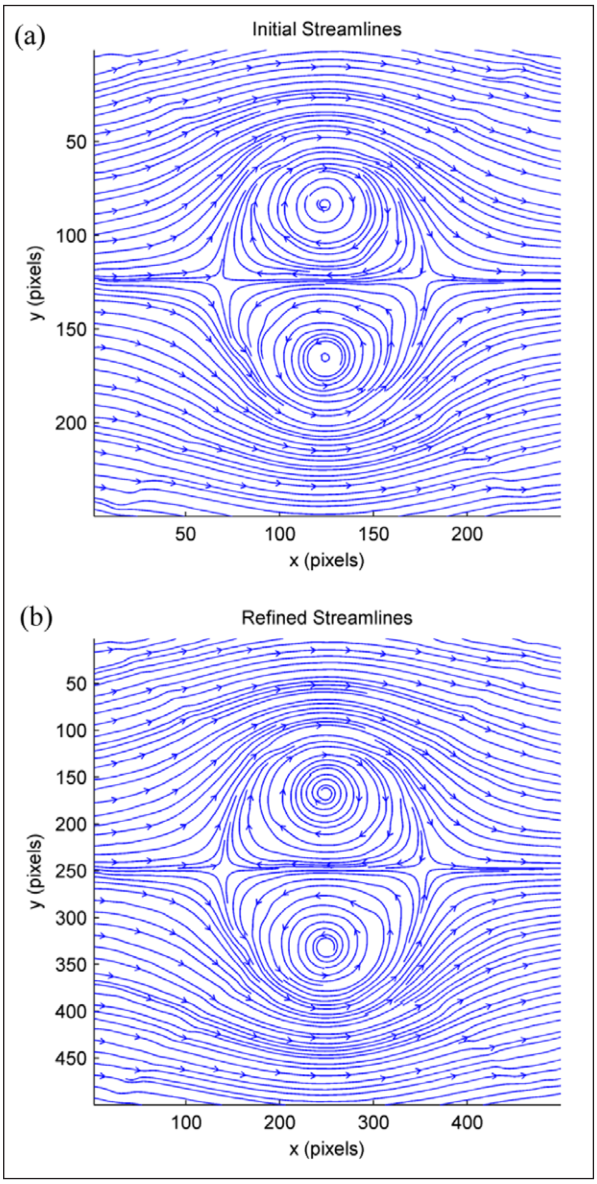

Figure 3: Streamlines of an Oseen vortex pair in uniform flow extracted from the particle images, **(a)** Coarse-grained field, and **(b)** Refined field.

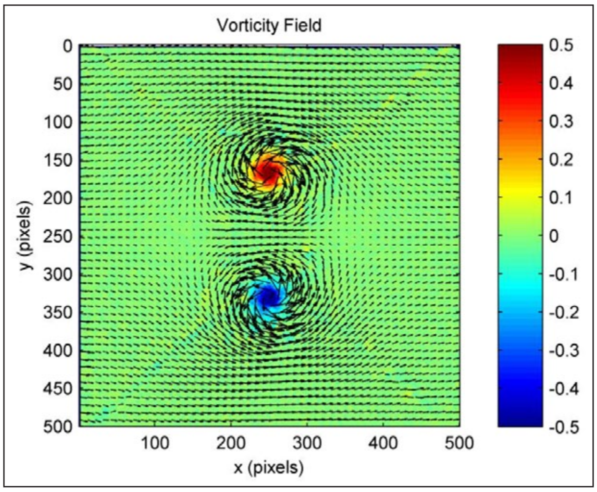

Figure 4: Vorticity field of an Oseen vortex pair in uniform flow extracted from the particle images.

simulations. It is found that the error is approximately proportional to the particle displacement and particle velocity gradient. In this case, the error is larger near the vortex cores where the velocity reaches the maximum and its gradient is very large. In these regions, the peak value of the root-mean-square (RMS) error in simulations is smaller than 0.4 pixels/unit-time, and the relative RMS error is about 2%. In general, the optical flow method can extract velocity fields with much higher spatial resolution and improved accuracy when the relevant parameters are suitably selected.

b. White Ovals on Jupiter

The White Ovals are distinct storms in the Jupiter's atmosphere. **Figure 5** shows two successive near-infrared continuum filtered (756 nm) images of the White Ovals. Three sets of images were taken on 02/19/1997 at a range of 1.1 million kilometers by the Solid State Imaging system aboard NASA's Galileo spacecraft. Each was taken one hour apart. Unlike discrete particle images, these images have continuous patterns that are particularly suitable for the application of the optical flow method. It is noticed that the illumination from the Sun was considerably changed in a local and non-uniform fashion. Thus, the correction for this change is made by using the subroutine "correction_illumination.m" before applying the optical flow method. **Figure 6a** and **b** show the coarse-grained and refined velocity vector fields extracted from the White Ovals images, respectively. **Figure 7** shows the corresponding streamlines. **Figure 8** shows the refined vorticity field normalized by its maximum value. It is indicated the interactions between the three cyclonic vortices. The balloon-shaped vortex is seen between the two well-formed ovals. In optical flow computation in this case and the following cases, the Lagrange multipliers in the Horn-Schunck and Liu-Shen estimators are set at 20 and 2000, respectively.

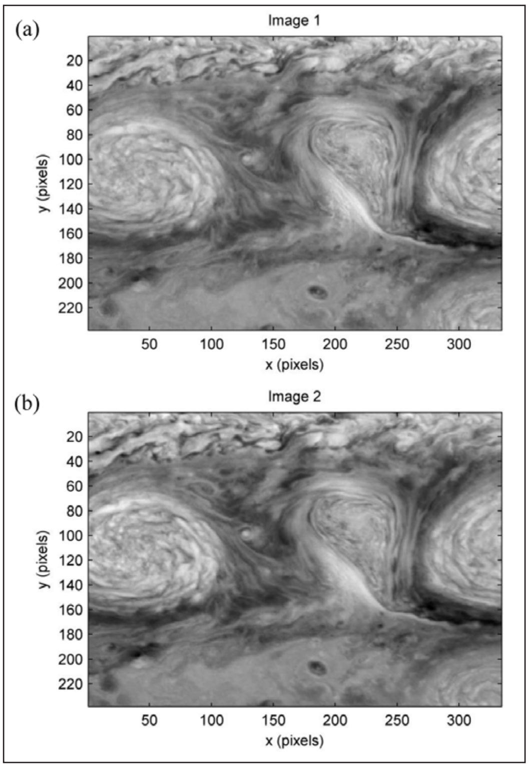

Figure 5: A pair of images of the White Ovals on Jupiter, **(a)** Image #1, and **(b)** Image #2.

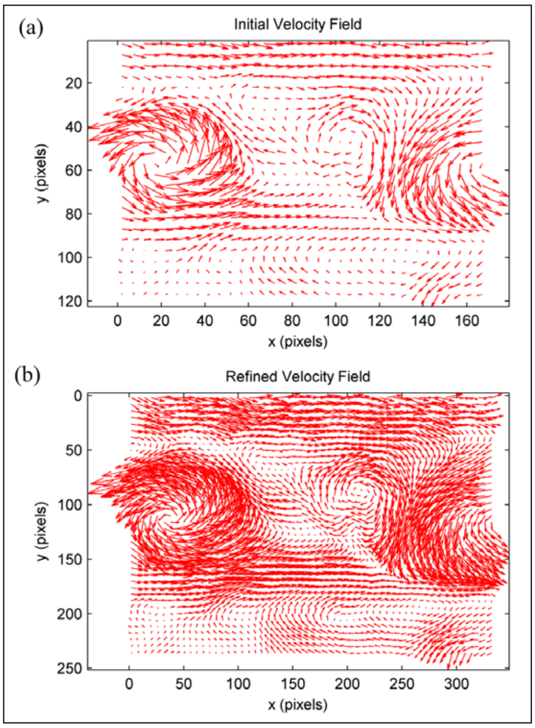

Figure 6: Velocity vectors of the White Ovals on Jupiter, **(a)** Coarse-grained field, and **(b)** Refined field.

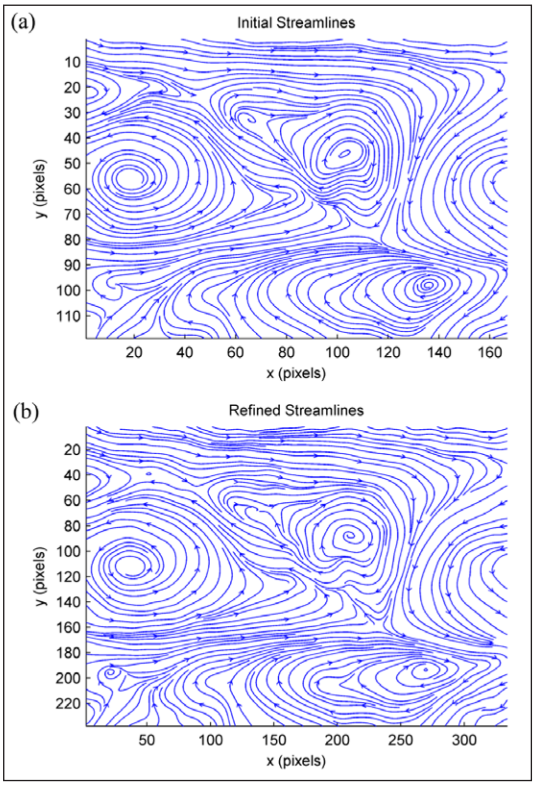

Figure 7: Streamlines of the White Ovals on Jupiter, **(a)** Coarse-grained field, and **(b)** Refined field.

c. Other examples

Quasi 2D turbulence is simulated by randomly distributing 50 Oseen vortices in the uniform distribution in a cloud image. The distribution of the strengths of the vortices is the Gaussian distribution with zero mean and the standard deviation of $\Gamma = 3000$ (pixel)^2/s. The sizes of the vortices are distributed in a Gaussian distribution with the mean of $r_0 = 20$ pixels and the standard deviation of 10 pixels. One of the pair of the cloud images is shown in **Figure 9a**. **Figure 9b** shows the velocity vectors overlaid on the normalized vorticity field extracted by the optical flow algorithm from the cloud image pair. The optical flow method can give better estimation in the distribution of velocity particularly in the regions with rapid changes in comparison with the traditional cross-correlation method applied to such images. The mean error in the whole image is about 0.03 pixels/unit-time in this case.

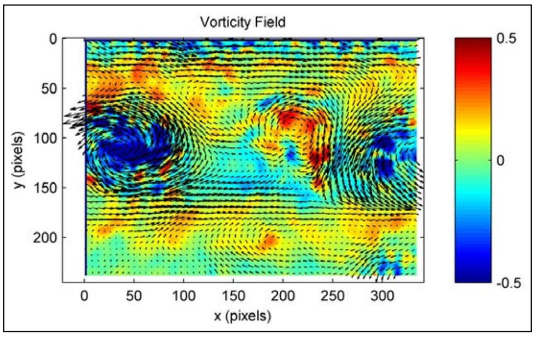

Figure 8: Vorticity field of the White Ovals on Jupiter.

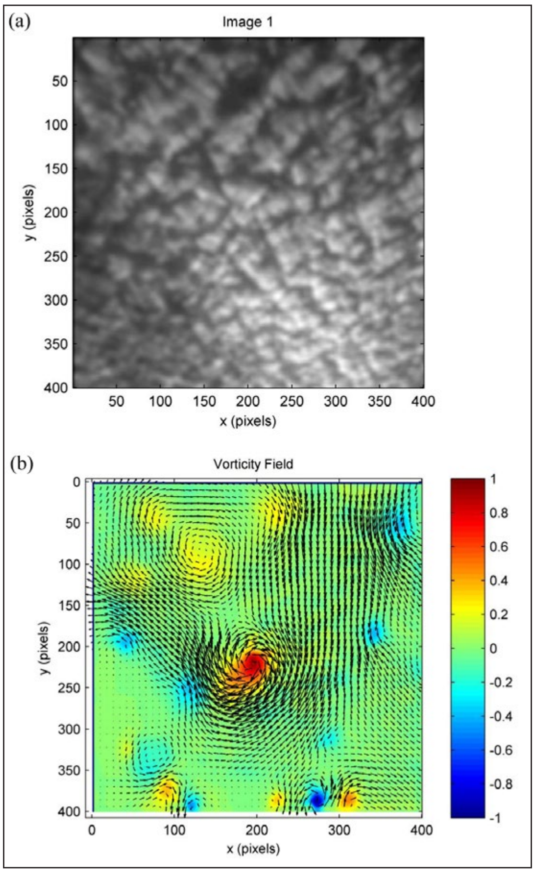

Figure 9: Quasi 2D turbulence, **(a)** Sample image, and **(b)** Vorticity field.

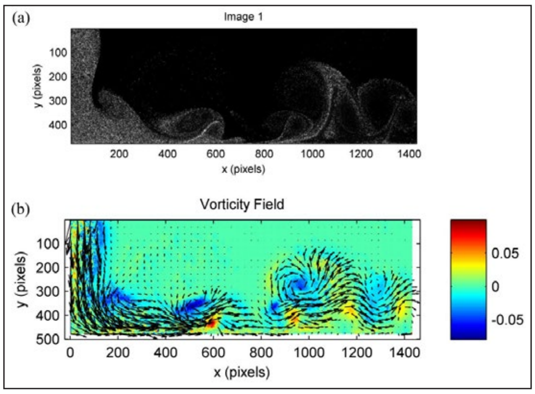

Figure 10: Wall-jet region of an impinging jet, **(a)** Sample image, and **(b)** Vorticity field.

PIV images were obtained in an air jet normally impinging on a wall from a contoured circular nozzle in experiments, focusing on the wall-jet region where strong interactions between vortices and boundary layer occur [11]. **Figure 10a** shows one sample PIV image of the wall-jet region with relatively high particle density. **Figure 10b** shows the velocity vectors overlaid on the normalized vorticity field. The optical flow method is able to provide the velocity field near the wall at the resolution of one vector per one pixel. The optical flow is able to reveals large vortices generated in the shear layer of the free jet and wall jet due to the Kelvin-Helmholtz instability, and induced secondary vortices (secondary separations) in boundary layer near the wall. Compared to the traditional cross-correlation method in PIV, the optical flow method yields the velocity field with much higher spatial resolution (one vector per a pixel), revealing more details of the near-wall flow structures in the thin wall jet (the thickness is about 3 mm in the physical scale, and 100 pixels in the image plane).

(2) Availability

Operating system

Based on Matlab (R2007a, or later versions): Windows.

Programming language

Matlab (R2007a, or later versions).

Additional system requirements

None.

Dependencies

Several functions in Matlab image processing toolbox are required.

List of contributors

Tianshu Liu, Department of Mechanical and Aerospace Engineering, Western Michigan University.

Lixin Shen, Department of Mathematics, Syracuse University.

Software location

Name: GitHub
Identifier: https://github.com/Tianshu-Liu/OpenOpticalFlow
License: MIT license
Data published: April 12, 2017

(3) Reuse Potential

Global velocity diagnostics is of fundamental importance in the study of fluid mechanics in order to understand the physics of complex flows. In addition to particle image velocimetry (PIV), the optical flow method allows extraction of high-resolution velocity fields from flow visualization images obtained in various measurements in different scientific and engineering fields. These images could be obtained by using inexpensive CCD and CMOS cameras under various illumination conditions. The physical and mathematical foundations of the optical flow method are well established. Optical flow computation can be conducted in a PC with Matlab to extract velocity fields at a spatial resolution of one vector per a pixel. The optical flow method has been used in some challenging flow measurements in which the traditional cross-correlation method in PIV could not provide sufficient high-resolution velocity fields in large domains [7–12]. Although there are research optical flow programs developed by different groups, an easy-to-use open source program is still lacking for non-expert users in various settings of fluid-mechanic applications. This open source program, OpenOpticalFlow, allows users to adapt the optical flow method for their specific problems. Furthermore, OpenOpticalFlow can be integrated with Matlab for convenient image processing, data presentations, and data input/output. Besides GitHub, the programs can be directly downloaded from the author's website https://wmich.edu/mechanical-aerospace/directory/liu.

Competing Interests

The author has no competing interests to declare.

References

1. **Quenot, G M, Pakleza, J** and **Kowalewski, T A** 1998 Particle image velocimetry with optical flow. *Exp Fluids*, 25: 177–189. DOI: https://doi.org/10.1007/s003480050222
2. **Corpetti, T, Memin, E** and **Perez, P** 2002 Dense estimation of fluid flows. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 24: 365–380. DOI: https://doi.org/10.1109/34.990137
3. **Corpetti, T, Heitz, D, Arroyo, G** and **Memin, E** 2006 Fluid experimental flow estimation based on an optical flow scheme. *Exp Fluids*, 40: 80–97. DOI: https://doi.org/10.1007/s00348-005-0048-y
4. **Ruhnau, P, Kohlberger, T, Schnorr, C** and **Nobach, H** 2005 Variational optical flow estimation for particle image velocimetry. *Exp Fluids*, 38: 21–32. DOI: https://doi.org/10.1007/s00348-004-0880-5
5. **Heitz, D, Héas, P, Mémin, E** and **Carlier, J** 2008 Dynamic consistent correlation-variational approach for robust optical flow estimation. *Exp Fluids*, 45: 595–608. DOI: https://doi.org/10.1007/s00348-008-0567-4
6. **Heitz, D, Memin, E** and **Schnorr, C** 2010 Variational fluid flow measurements from image sequences: synopsis and perspectives. *Exp Fluids*, 48: 369–393. DOI: https://doi.org/10.1007/s00348-009-0778-3
7. **Liu, T** and **Shen, L** 2008 Fluid flow and optical flow. *J Fluid Mech*, 614: 253–291. DOI: https://doi.org/10.1017/S0022112008003273
8. **Liu, T, Wang, B** and **Choi, D** 2012 Flow structures of Jupiter's Great Red Spot extracted by using optical flow method. *Phy Fluids*, 24: 096601–13.
9. **Liu, T, Nink, J, Merati, P, Tian, T, Li, Y** and **Shieh, T** 2010 Deposition of micron liquid droplets on wall in impinging turbulent air jet. *Exp Fluids*, 48: 1037–1057. DOI: https://doi.org/10.1007/s00348-009-0790-7
10. **Hayasaka, K, Tagawa, Y, Liu, T** and **Kameda, M** 2016 Optical-flow-based background-oriented schlieren technique for measuring a laser-induced underwater shock wave. *Exp Fluids*, 57: 179. DOI: https://doi.org/10.1007/s00348-016-2271-0
11. **Liu, T, Merat, A, Makhmalbaf, M H M, Fajardo, C** and **Merati, P** 2015 Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. *Exp Fluids*, 56: 166–189. DOI: https://doi.org/10.1007/s00348-015-2036-1
12. **Wang, B, Cai, Z, Shen, L** and **Liu, T** 2015 An analysis of physics-based optical flow method. *J Comp Appl Math*, 276: 62–80.
13. **Horn, B K** and **Schunck, B G** 1981 Determining optical flow. *Artificial Intelligence*, 17: 185–204. DOI: https://doi.org/10.1016/0004-3702(81)90024-2

How to cite this article: Liu, T 2017 OpenOpticalFlow: An Open Source Program for Extraction of Velocity Fields from Flow Visualization Images. *Journal of Open Research Software*, 5: 29, DOI: https://doi.org/10.5334/jors.168

Submitted: 07 March 2017 **Accepted:** 29 September 2017 **Published:** 20 October 2017

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS