SoftwareX 12 (2020) 100537

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

piv-image-generator: An image generating software package for planar PIV and Optical Flow benchmarking

Luís Mendes [a,*], Alexandre Bernardino [b], Rui M.L. Ferreira [a]

[a] CERIS, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal
[b] ISR, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 7 February 2020
Received in revised form 4 June 2020
Accepted 4 June 2020

Keywords:
PIV
Optical flow
Hydraulics
Fluid mechanics
Velocimetry

ABSTRACT

We describe a generator of synthetic images of tracers in turbulent flows to benchmark Particle Image Velocimetry and Optical Flow algorithms. Generated flows include: uniform, shear flows, inviscid stagnation point flows, and decaying and Rankine vortices. Image control parameters include particle and illumination characteristics, noise level and image bit depth, among others. The software package is unique in the sense that image characteristics and error sources are fully parameterized and that it provides zero-uncertainty ground truth, a combination of features never before conjoined in synthetic particle image generators. Examples of application are included in the text.

Code metadata

Current code version	v1.1.2
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_33
Code Ocean compute capsule	
Legal Code License	GNU General Public License v2
Code versioning system used	git
Software code languages, tools, and services used	MATLAB
Compilation requirements, operating environments & dependencies	MATLAB for Linux, OS X or Windows
If available Link to developer documentation/manual	A Technical manual PDF is provided in the submission
Support email for questions	luis.mendes@tecnico.ulisboa.pt

Software metadata

Current software version	v1.1.2
Permanent link to executables of this version	https://git.qoto.org/CoreRasurae/piv-image-generator/-/archive/v1.1.2/piv-image-generator-v1.1.2.zip
Legal Software License	GNU General Public License v2
Computing platforms/Operating Systems	MATLAB for Linux, OS X or Microsoft Windows
Installation requirements & dependencies	MATLAB
If available, link to user manual - if formally published include a reference to the publication in the reference list	
Support email for questions	luis.mendes@tecnico.ulisboa.pt

1. Motivation and significance

Planar Particle Image Velocimetry (PIV) is an image processing method supported by cross correlation, which is able to extract two component velocity vectors in a two dimensional grid from a pair of tracer images taken with a fixed time interval. Tracer images are black and white images of bright spots against a dark

* Corresponding author.
E-mail addresses: luis.mendes@tecnico.ulisboa.pt (L. Mendes), alex@isr.tecnico.ulisboa.pt (A. Bernardino), ruimferreira@tecnico.ulisboa.pt (R.M.L. Ferreira).

https://doi.org/10.1016/j.softx.2020.100537

background resulting from light reflected from tracer particles when moving across an illuminated light sheet. These are artificially inserted in the medium of interest (e.g. fluid, air, etc.). The tracer particles are suspended in the medium, are small and ideally have a density close to that of the medium, so that their movement is highly related with the flow or the medium. The grid is normally a matrix of contiguous regular elements, called interrogation areas (IAs). A two component velocity vector is obtained for each IA, for each image pair.

PIV is widely used in industrial applications and in scientific explorations, (for example, see [1]). Example applications include fluid mechanics, in general [2], aerodynamics, e.g. aircraft and airfoil profile design [3], characterization of building and bridge aerodynamic actions [4], sports science [5], combustion, e.g. spray optimization in internal engines [6], hydraulics, e.g. environmental flows [7] or bio fluid mechanics e.g. vascular mechanics [8]. Optical Flow (OF) methods are widely used in the robotics field [9], computer vision [10], autonomous vehicles [11], but also, for replacing or complementing PIV cross-correlation usage cases [12].

Databases of tracer based images that include the ground truth to validate velocimetry instrumentation in fluid applications are not common. The few relevant examples, (for example, see [13,14]), have provided valuable means to assess the quality of PIV and Optical Flow algorithms [15]. However, to the best of our knowledge, we find that there are no databases or tools to generate images that simultaneously: (i) include rotational and distortional flows and combinations of both as well as flows with singularities; (ii) provide the ground truth based on theoretical solutions and thus with no uncertainty or, at least, low and quantifiable uncertainty; (iii) include the main causes for error in PIV measurements as parameters that can be manipulated by the user when generating the images (the latter include size of particles, out-of-plane movement, white noise and particle-related noise, tracer density, thickness of the light sheet, image bit depth, number of repetitions with same parameters — for statistical relevance); (iv) allow for the precise specification of the number of tracer particles in the image and the number of particles whose illumination is below the threshold (and are considered noise), the reference maximum displacement in a ROI (region of interest), and the stochastic control of out-of-plane motion.

We aim at bridging this knowledge gap by creating piv-image-generator that generates such databases from a set of user-defined relevant control parameters including tracer image size, particle concentration, out-of-plane motion, white noise level or pixel bit depth.

The images are relative to different user-defined flow types. The software package can be easily extended to support additional types, should users identify extra needs. Besides generating test images, the ground truth velocity is also exported by piv-image-generator. A single MATLAB data file is created that contains validation velocities for PIV methods (generated at the center of IAs) as well as validation velocities for dense Optical Flow (OF) methods (generated at the centers of pixels).

The piv-image-generator can be used to provide test images pairs and reference data for performing comparison and benchmarking of PIV and OF methods regarding their overall and spatially specific accuracy, for the different control parameters that the image generator supports. We provide, in the software repository, an example for sample tracer based image generation [16].

The paper is structured around four other sections. Section "Software description" provides the software package fundamentals and gives a general overview. Section "Illustrative Examples" provides some examples for software use cases. Section "Impact" sketches how the software package contributed to research developments and its expected near future influence for research, as well as, for the society. Closing the paper, section "Conclusions" presents a final discussion and lines out possible future work.

2. Software description

2.1. General description

The software processing phases can be summarized in the following seven steps:

1 The user defines the image generation parameters
2 A planar flow field is generated according to the user preferences
3 Particles are randomly placed, including the lateral out of plane initial position, in the first image
4 Particles are displaced according to the planar flow field in the second image and according to the lateral (out of plane) movement
5 Images are rendered
6 Noise perturbations are superimposed to the image
7 Images are exported to file in TIF format, along with validation data file in MATLAB format

2.2. Flow types

Tracer images are black and white images of bright spots against a dark background. Resulting from light reflected from tracer particles when moving across an illuminated light sheet. These are artificially inserted in the medium of interest (e.g. fluid, air, etc.). Are in suspension in the medium, are small and ideally have a density close to that of the medium so that their movement is highly related with the flow or the medium.

The piv-image-generator includes the following flows:

1 uniform flow;
2 Poiseuille flow;
3 simple shear flow;
4 inviscid stagnation point flow [17];
5 Rankine vortex;
6 Rankine vortex with superimposed uniform flow, and
7 decaying vortex (similar to a Lamb–Oseen vortex, as proposed by [14]).

Displacement fields are found by analytically integrating the corresponding velocity field in all but one case, the Rankine vortex with superimposed uniform flow. In the latter case, a numerical integration with a 4th-order Runge–Kutta method is employed. The uncertainty in particle position is zero in all cases but #6; in the case of flow #6, the error in particle positioning can be neglected as is $O(\Delta t)^3$ and the time interval, Δt, between images, is small for both PIV and OF applications.

Table 1 details the equations that describe the above flows and corresponding displacement fields, as implemented in piv-image-generator. In these equations, x and y are Cartesian orthogonal spatial coordinates specifying a plane, x_0 and y_0 are initial positions, r is the radial polar coordinate, θ is the azimuthal polar coordinate, θ_0 is an initial azimuthal position, t is the time elapsed since an initial instant, x_c and y_c are the coordinates of the zero-velocity locus, in the case of the Shear Flow, or the coordinates of the stagnation point, for the inviscid stagnation point flow, or the center of the vortex, for both Rankine and Decaying vortices; y_{max} and y_{min} are the coordinates of the upper and lower bounds of the flux tube within which the Poiseuille flow takes place; R is the radius for which maximum azimuthal velocity is attained in the decaying and Rankine vortex flows, Γ is the circulation of rotational core, in the Rankine vortex, m is a reference velocity, M is a scaling parameter that allows for the specification of the desired maximum relative displacement (a percentage of the IA size, always less than 25%). For the Rankine Vortex and Rankine with superimposed Uniform flow cases $\Gamma = m2\pi R$. For the

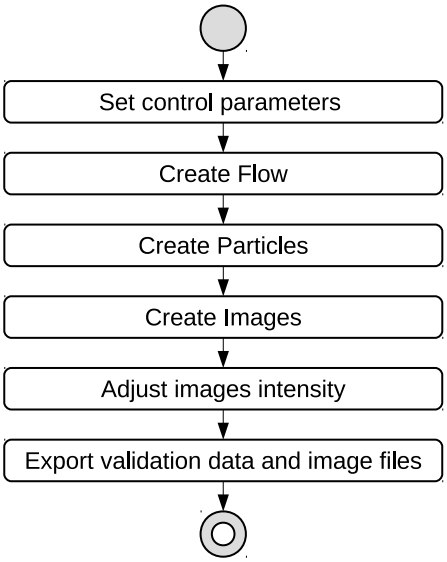


Fig. 1. Flowchart diagram of PIV image generator software package.

decaying vortex, $\Gamma = m2\pi R$ and $\alpha = 1.25643$ are parameters that express the vortex core circulation, as specified by [14].

This flow field parameterization is entirely independent of PIV or Optical flow image variables, such as the size of the IA, as m is an actual velocity, expressed in m/s. In the case of PIV applications, however, it is intended that the maximum displacement is a fraction of the size of the PIV IA. This is achieved by specifying appropriate values of M to transform the velocity from physical coordinates to image coordinates per unit time. In the parabolic flow, for instance, the scaling parameter M is:

$$M = (\frac{y_{max} - y_{min}}{2} - y_{min})(\frac{y_{max} - y_{min}}{2} - y_{max}).$$

The value of the time stamp of the second image, t, is computed as $t = \frac{M}{m} D_{max}$ where D_{max} is the maximum displacement (in pixels). In the case of PIV applications, $D_{max} = pI_A$ where p denotes the percentage of the size of the PIV interrogation area I_A (in px).

The set of flows included in piv-image-generator is adequate and sufficient for the purpose of testing planar PIV and Optical Flow algorithms. In continuum mechanics, all velocity gradients can be decomposed into a distortional component and a rotational component. The set of flows included in piv-image-generator represent both distortional and rotational flows as well as singularities and simple zero gradient flows. The uniform flow, for which the velocity gradient is zero, is the simplest case. The shear and Poiseuille flows are simultaneously rotational and distortional, the decaying vortex and the core of the Rankine vortex are purely rotational, and the axisymmetric stagnation point flow and the outer ring of the Rankine vortex are purely distortional. Additionally, the superposition of a uniform flow to a Rankine vortex is included as an example of a flow for which the streamlines cannot be used to base criteria to detect rotational motion.

2.3. Software architecture

The main software functionality is to generate synthetic PIV images that mimic true PIV images obtained in laboratory environments, while having zero-uncertainty ground truth. The images and underlying flow fields are generated according to the flow diagram presented in Fig. 1.

The user defines the control parameters for the synthetic generator in the application entry files. The piv-image-generator application then combines them with pre-defined expressions or constants to derive the configuration input parameters for the supporting library. Further details about those internal expressions and constants can be found in the technical manual. The Create Flow step then instantiates the proper flow model and generates the flow field from it. With the instantiated flow field and control parameters, a number of particles are generated according to the N_i concentration and uniformly randomly distributed across each IA, both in-plane wise, as out-of-plane wise. Displacements are then computed for the out-of-plane movement. Finally, in Create Images step, particles are rendered into an image pair, according to peak intensity, image resolution and flow field displacement. White Gaussian Image Noise (WGIN) is superimposed to the images. Peak image intensity is either clipped, or normalized, depending on user preferences. Finally the image bit depth resolution is enforced in `AdjustImagesIntensity` function. The last step then exports the validation data into a MAT file along with the image pair, that is exported in TIF format.

For further implementation details the reader may consult the software package technical manual and the source code itself.

2.4. Image parameterization and generation

The piv-image-generator generates pairs of images of tracers illuminated by a Gaussian laser sheet. All particles are characterized by the coordinates of their centers in the 3D Cartesian space. The projection of these coordinates in the x, y plane is randomly generated in the first image. The lateral coordinate, i.e. the out-of-plane motion, is also randomly generated for the first image. The user decides how many particles should be visible in each interrogation area by choosing the value of N_i, the number particle concentration (number of illuminated particles per PIV interrogation area). The size of the interrogation area, I_A, must be previously chosen. The total number of generated particles is however larger, with some particles placed beyond the threshold I_0/e^2, where e is the Euler's number, of the laser sheet intensity, where I_0 is the maximum intensity. The thickness of the laser sheet (out-of-plane distance encompassing the region where laser intensity is larger than I_0/e^2), its standard variation and the maximum intensity are susceptible to be modified by the user.

For each flow type identified in Section 2.2, the centers of each particle in the x, y coordinates are displaced as per the equations shown in Table 1. So, in the initial image, the number of illuminated particles in each IA is exactly N_i, but that may not be the case for the second image. The lateral coordinate, is randomly generated to simulate the effect of turbulence. The turbulence intensity is defined by the user as the standard deviation of the out-of-plane velocity fluctuations (`outOfPlaneStdDeviation`) (see Table 2). Some particles may thus leave or enter the laser sheet, thus resulting in a total number of illuminated particles in the second image different from that in the first image. Note that the expected value of the difference in the initial and final N_i is zero since the out-of-plane motion is Gaussian with zero mean.

Each image spot, i.e., the imaged particle, has its size defined in pixels(px) and the image spot intensities are modeled as a Gaussian distribution of gray levels, essentially following the procedure of [18]. The user may choose the diameter of the image spot, from its fractional radius size ($2.0\times$`particleRadius`). The actual aspect of the spot depends on the position of its center in the (x, y) plane and its lateral position — the light reflected by each particle depends on the incident light which, in turn, depends on its lateral position. A particle thus appears smaller if it is not exactly in the center of the laser sheet. Note also that particles that are beyond the I_0/e^2 still reflect light, but are so dim that its contribution is considered noise.

Table 1
Flow types, velocity flow fields and displacement equations.

Flow type	Displacements in the flow field	Velocity flow field
Uniform	$\begin{cases} x_1 = \frac{m}{M}t + x_0 \\ y_1 = y_0 \end{cases}$	$\begin{cases} u = \frac{m}{M} \\ v = 0 \end{cases}$
Poiseuille	$\begin{cases} x_1 = \frac{m}{M}(y_0 - y_{min})(y_0 - y_{max})t + x_0 \\ y_1 = y_0 \end{cases}$	$\begin{cases} u = \frac{m}{M}(y - y_{min})(y - y_{max}) \\ v = 0 \end{cases}$
Stagnation	$\begin{cases} x_1 = e^{\frac{m}{M}{}_x t}(x_0 - x_c) + x_c \\ y_1 = e^{-\frac{m}{M}{}_y t}(y_0 - y_c) + y_c \end{cases}$	$\begin{cases} u = \frac{m}{M}{}_x(x - x_c) \\ v = -\frac{m}{M}{}_y(y - y_c) \end{cases}$
Simple shear	$\begin{cases} x_1 = \frac{m}{M}y_0 t + x_0 \\ y_1 = y_0 \end{cases}$	$\begin{cases} u = \frac{m}{M}y \\ v = 0 \end{cases}$
Shear rotated by θ	$\begin{cases} x_1 = \frac{m}{M}\left((y_0 - y_c)\cos(\theta) + \right. \\ \quad \left. + (x_0 - x_c)\sin(\theta)\right)\cos(\theta)t + x_0 \\ y_1 = -\frac{m}{M}\left((y_0 - y_c)\cos(\theta) + \right. \\ \quad \left. + (x_0 - x_c)\sin(\theta)\right)\sin(\theta)t + y_0 \end{cases}$	$\begin{cases} u = \frac{m}{M}\left((y - y_c)\cos(\theta) + \right. \\ \quad \left. + (x - x_c)\sin(\theta)\right)\cos(\theta) \\ v = -\frac{m}{M}\left((y - y_c)\cos(\theta) + \right. \\ \quad \left. + (x - x_c)\sin(\theta)\right)\sin(\theta) \end{cases}$
Rankine	$\begin{cases} x_1 = r\cos(\frac{\Gamma}{2\pi R}\frac{1}{MR}t + \theta_0) \\ \qquad\qquad , r \le R \\ y_1 = r\sin(\frac{\Gamma}{2\pi R}\frac{1}{MR}t + \theta_0) \end{cases}$ $\begin{cases} x_1 = r\cos(\frac{\Gamma}{2\pi R}\frac{R}{Mr^2}t + \theta_0) \\ \qquad\qquad , r > R \\ y_1 = r\sin(\frac{\Gamma}{2\pi R}\frac{R}{Mr^2}t + \theta_0) \end{cases}$	$u_\theta(r) = \begin{cases} \frac{\Gamma}{2\pi R}\frac{r}{MR} & , r \le R \\ \frac{\Gamma}{2\pi R}\frac{R}{Mr} & , r > R \end{cases}$
Decaying	$\begin{cases} x_1 = r\cos(\omega_v(t, \theta_0, r)) \\ y_1 = r\sin(\omega_v(t, \theta_0, r)) \\ \omega_v(t, \theta_0, r) = \frac{\Gamma}{2\pi R}(\frac{1}{r} + \frac{1}{2\alpha}\frac{R}{r^2}) \times \\ \quad \times \frac{1}{M}\left(1 - e^{(-\alpha\frac{r^2}{R^2})}\right)t + \theta_0 \end{cases}$	$u_\theta(r) = \frac{\Gamma}{2\pi R}\left(1 + \frac{1}{2\alpha}\frac{R}{r}\right) \times \quad \frac{1}{M}\left(1 - e^{(-\alpha\frac{r^2}{R^2})}\right)$
Uniform and Rankine	Numerically solved by 4th order Runge–Kutta	$\begin{cases} u = -\frac{\Gamma}{2\pi R}\frac{y}{MR} + u_0 \\ \qquad\qquad , r \le R \\ v = \frac{\Gamma}{2\pi R}\frac{x}{MR} + v_0 \end{cases}$ $\begin{cases} u = -\frac{\Gamma}{2\pi R}R\frac{y}{M(x^2+y^2)} + u_0 \\ \qquad\qquad , r > R \\ v = \frac{\Gamma}{2\pi R}R\frac{x}{M(x^2+y^2)} + v_0 \end{cases}$

Each image is lossless TIF with 8 or 16 bit depth gray scale images, where only 8, 10 or 12 bits are used to encode the tracer image simulating the light intensity reflected by particles of specified diameters (see `bitDepths` description in Section 3).

White Gaussian Image Noise (WGIN) is added to both images independently (`noiseLevel`). The synthetic images thus generated are very similar to true planar laser-illuminated tracer images obtained in laboratory environments, as seen in Figs. 3a, 5a and 5b. Note however, that all image features are decided by the user and correspond to a zero-uncertainty ground truth. This configures an absolute novelty in what concerns synthetic image generators as the performance of evaluated algorithms can be directly assessed as a function of image parameters for which there is no uncertainty.

The application is supported by an internal library, so that the application parameters are translated into library parameters, while others are directly set by the application. Table 2 summarizes the parameters that can be set by the user in the application, as well as, the parameters that the library uses, but are not exposed by the application to the user. Should the end-user intend to create a custom application, then all library parameters are accessible, and they are documented in the technical manual.

3. Illustrative examples

In this section, examples are provided for some parameters.

1. Fig. 2a depicts an uniform flow field
2. Fig. 2b refers to a Poiseuille flow field
3. Fig. 2c represents the flow field for the Rankine vortex.
4. Fig. 2d depicts the Rankine vortex with superimposed uniform flow field
5. Fig. 2e shows an inviscid stagnation point flow field.

These are examples of, some of, the currently supported flow fields as implemented by piv-image-generator.

Particle size

Fig. 3a shows a sample synthetic PIV image with particle diameter of 3 px (pixels) and a number particle concentration by volume set to $N_i = 6$, while Fig. 3b shows a sample synthetic PIV image with particle diameter of 6 px and $N_i = 6$. Both images are shown as generated by the piv-image-generator. Fig. 5a depicts a synthetic PIV image with $N_i = 12$, while Fig. 5b presents a

Table 2
Configurable control parameters.

Application input parameters	
`displayFlowField`	Flag for optional generation of flow field preview
`closeFlowField`	Flag for automated closing of flow field figure
`sizeX`	Target image horizontal resolution
`sizeY`	Target image vertical resolution
`bitDepths`	Image bit depth
`flows`	List of flow type to generate
`deltaXFactor`	Displacement ratio relative to the IA size
`particleRadius`	The particle radius in *px*
`Ni`	Number particle concentration
`noiseLevel`	Noise Level in dBW
`outOfPlaneStdDeviation`	Out of plane standard deviation (mm)
`numberOfRuns`	Number of runs (details in Section 3)
Library constant input parameters	
`flowParameters.maxVelocity`	Maximum velocity in mm/s
`pivParameters.lastWindow`	IA size for last window [px , px]
`pivParameters.laserSheetThickness`	Laser sheet thickness (mm)
`imageProperties.mmPerPixel`	Pixel to millimeters scale
`pivParameters.particleIntensityPeak`	Peak reflected light intensity

Listing 1: "exampleAllTestImagesMain.m"

```
sizeX=512; %Image width without margins
sizeY=512; %Image height without margins
flows={'uniform' 'parabolic' 'stagnation' 'shear' ...
    'shear_45d0' 'shear_22d3' 'rankine_vortex' ...
    'decaying_vortex' 'rk_uniform'};
bitDepths=8;
deltaXFactor=[0.05 0.25];
particleRadius=[1.5 3.0];
Ni=[6 12];
noiseLevel=[0 15];
outOfPlaneStdDeviation=[0.025 0.10];
numberOfRuns=2;

generatePIVImagesWithAllParametersCombinations;
```

synthetic PIV image with $N_i = 16$, both with particle diameters of 4 px and WGIN equal to 10 dBW.

Fig. 4a shows a sample synthetic PIV image with $N_i = 6$, while Fig. 4b shows a sample synthetic PIV image with $N_i = 12$. Both images have particle diameter of 3 px (pixels) and are shown as generated by piv-image-generator.

User code sample

Listing 1 is a slightly modified, but functional version taken from `exampleAllTestImagesMain.m` in [16].

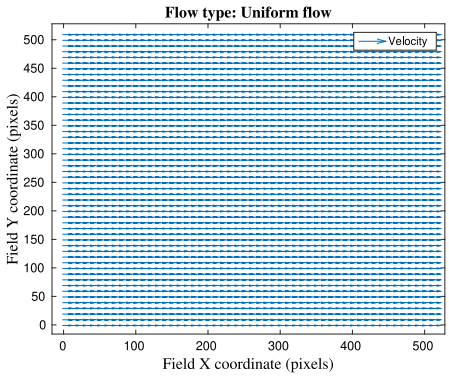
This configuration will generate images with 512 pixels × 512 pixels for each exhaustive combination of the parameters. This means that, e.g. Uniform flow will be generated for the image bit depth of 8 bits (because is the only value specified) and for each value of the displacement parameter `deltaXFactor`, combined with each value of the particle radius parameter `particleRadius` and so on. Upon exhausting the remaining combinations, the next flow is considered, until all flow options are exhausted.

1. `bitDepths` can be either 8, 10 or 12-bits and defines the pixel bit depth for the generated images. The pixel bit depth will affect the gray scale used to represent the brightness. Thus, the 8-bit grayscale will be in the range of 0 to 255 brightness values and the 8-bit TIF file format will be used to store the images. For 10-bit bit depths, the brightness values will range from 0 to 1023, and the 16-bit TIF format will be employed. The 12-bit bit-depth will have brightness values in the range of 0–4095, using 16-bit TIF images. For the 10-bit and 12-bit bit depths, where the 16-bit TIF are employed, it also implies that images will look darker, because the file format brightness/luminosity ranges from 0 (Black) to 65 535 (White), but the generated images will have a much lower range, thus generic image viewer applications, that are not aware of the intended use case, will display dark images. If higher bit depths are needed, the code can be modified to support up to 16-bit, the current limit of TIF format.
2. `particleRadius` is the seeding particle radius in pixels.
3. The `deltaXFactor` defines in the maximum displacement as a proportion between the maximum displacement and the IA size. For instance, an IA of 16 × 16 with `deltaXFactor=0.25` will define a maximum velocity vector with magnitude 4.
4. The `noiseLevel` parameter specifies the WGIN (dB) values to consider. Like all the other parameters, can be a single value or an array of values of interest.
5. The `outOfPlaneStdDeviation` parameter defines the out of plane standard deviation and is related to the parameter `numberOfRuns`, in the sense that numberOfRuns should be greater or equal to the number of elements in `outOfPlaneStdDeviation`. For instance `numberOfRuns=2` will result in the generation of a single case of `outOfPlaneStdDeviation=0.025` and a single case of `outOfPlaneStdDeviation=0.10`. However if `numberOfRuns=7` then this would result in the generation of four images with `outOfPlaneStd- Deviation=0.025` and three images with parameter `outOfPlaneStd- Deviation=0.10`.

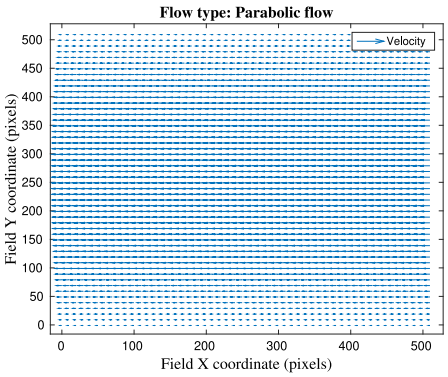
For the above listing, it generates 9 flows × 1 bit-depth × 2 displacement factors × 2 particle radius × 2 particle concentrations (Ni) × 2 noise-levels × 2 out of plane values, thus resulting in 288 total combinations, producing 288 synthetic PIV image pairs along with corresponding validation data.

4. Impact

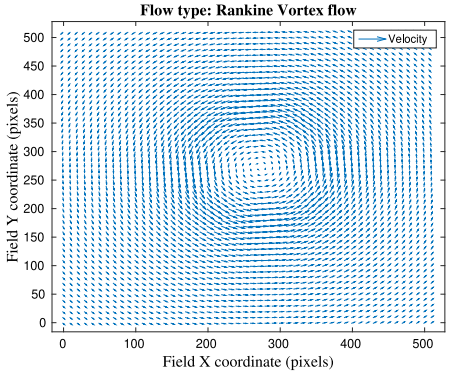
The piv-image-generator software package we have developed may be employed for a wide range of comparative studies



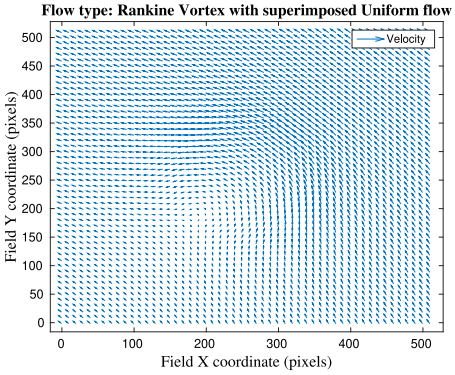
(a) Theoretical Uniform flow field in image coordinates (pixels)



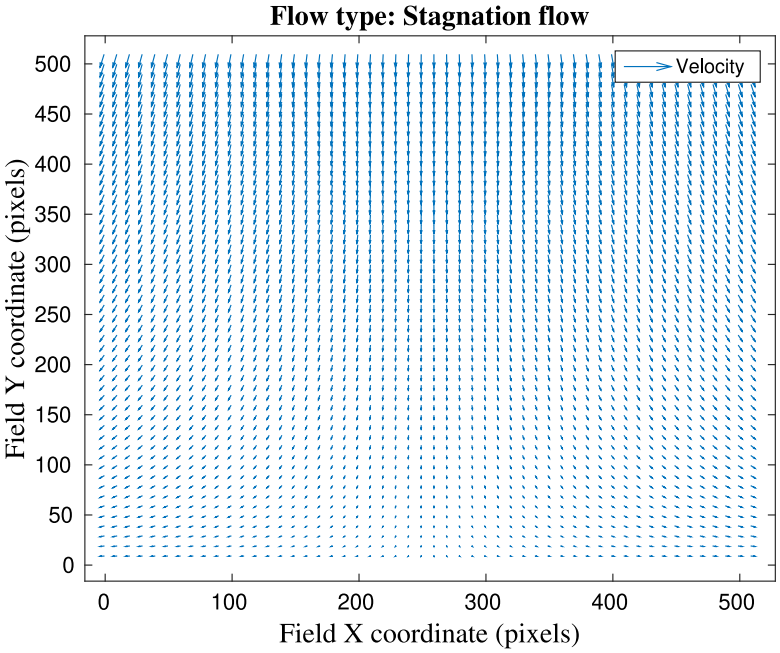
(b) Theoretical Parabolic flow field in image coordinates (pixels)



(c) Theoretical Rankine vortex flow field in image coordinates (pixels)



(d) Theoretical Rankine vortex with superimposed uniform flow field in image coordinates (pixels)

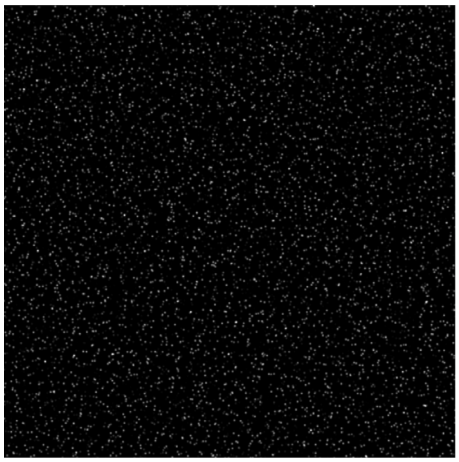


(e) Theoretical flow field with stagnation point in image coordinates (pixels)

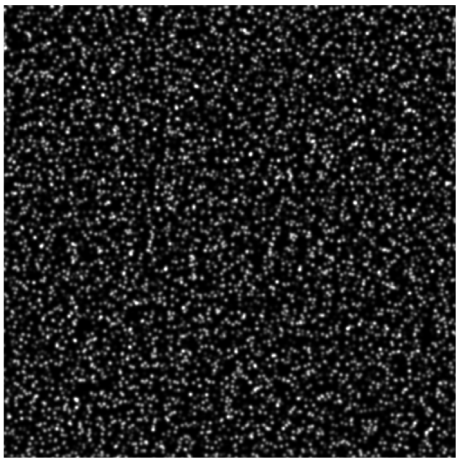
Fig. 2. Illustration of a subset of piv-image-generator supported flow types.

of PIV and OF algorithms. It may help in standardizing source data sets for benchmarking the methods under test. Different research groups may use it to compare their results allowing direct cross-validation efforts.

The set of flow types can be easily extended to support additional flow types, by defining additional flow objects that abide to the defined methods signatures. Other possible extensions include importing displacements maps from simulation tools, like DNS-generated flow fields, from which images with different particle image parameters can be created. The image generator can also be adapted for other fields of applications as it can be controlled to create the appropriate data sets, rather than limiting to a single generic data set.

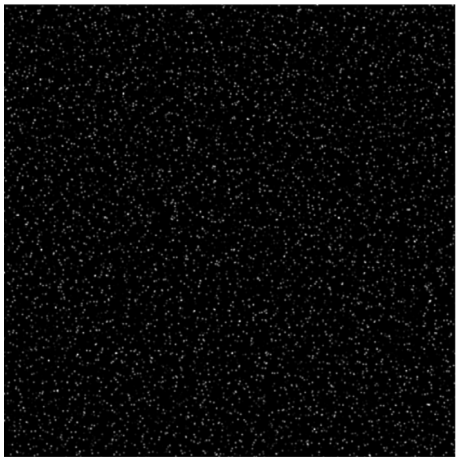


(a) Synthetic PIV image as generated by the piv-image-generator having $N_i = 6$ and 3 px diameter particles

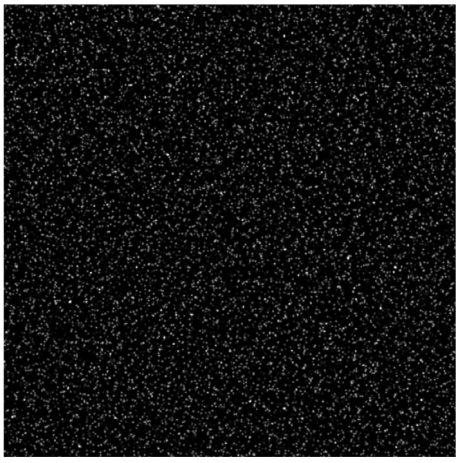


(b) Synthetic PIV image as generated by the piv-image-generator having $N_i = 6$ and 6 px diameter particles

Fig. 3. (a) and (b) illustrate the effect of having different particles sizes, while keeping the particle concentration constant.

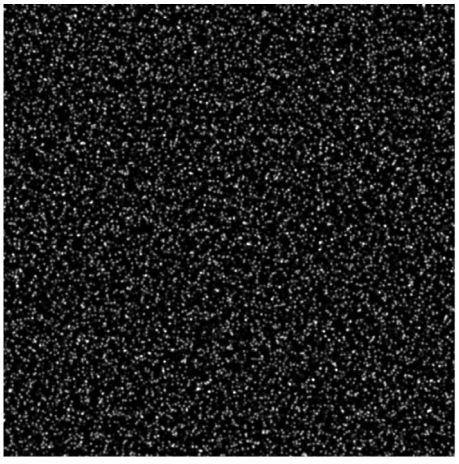


(a) Synthetic PIV image as generated by the piv-image-generator having $N_i = 6$ and 3 px diameter particles

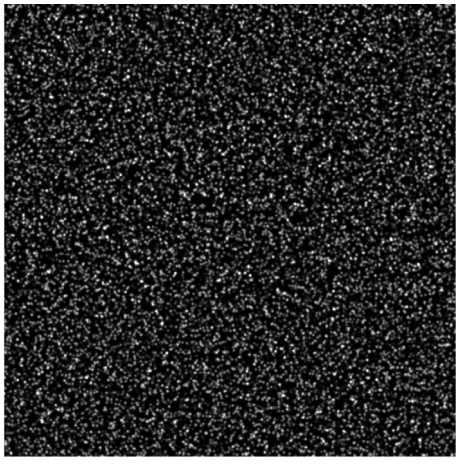


(b) Synthetic PIV image as generated by the piv-image-generator having $N_i = 12$ and 3 px diameter particles

Fig. 4. The sub-figures (a) and (b) illustrate the effect of different particle concentrations (N_i) for the same particle sizes.



(a) Synthetic PIV image as generated by the piv-image-generator with $Ni = 12$, WGIN 10 dBW and 4 px particles



(b) Synthetic PIV image as generated by the piv-image-generator having $N_i = 16$, 10 dBW WGIN and 4 px particles

Fig. 5. Sub-figures (a) and (b) illustrate the effect of different particle concentrations under the same White Gaussian Image Noise level of 10 dBW.

The piv-image-generator has already been used in a comparative study of Optical Flow and PIV methods, [19,20], where the image generator software package was used to generate a total of 7200 different synthetic PIV images, pertaining to 2160 different control parameter combinations, thus providing the test data for such comparative study. The most adequate particle sizes, in the sense that minimize velocity errors, and image properties where devised for each method. This example is representative but does not exhaust the possibilities of the piv-image-generator.

A non-exhaustive list of challenges posed to PIV and Optical Flow algorithms are summarized next.

1. Identifying the correct deformation rates for different orientations of the shear flow, namely when the axes of the image are not aligned with the principal deformation axes nor with the referentials for which one velocity component is null. This might highlight problems in PIV algorithms associated with peak locking due to inadequate modeling of sub-pixel motion.
2. Reproducing the theoretical Poiseuille profile independently of the orientation of the flow. This test may be useful to detect peak locking bias and susceptibility to insufficient tracers in regions with strong gradients (near the wall).
3. Identifying the stagnation point in the axisymmetric stagnation point flow. This is a relevant test to assess the response of the algorithms as the tracer density is reduced.
4. Guarantying zero vorticity in the stagnation point flow. This may be relevant to assess the susceptibility of the algorithms to out-of-plane loss of pairs that may generate erroneous vectors leading to apparent vortical motion.
5. Identifying the rotational core in the Rankine vortex and in the Rankine vortex with superimposed uniform flow. In the former flow, streamlines are circular but the flow is rotational in the core region only, which cannot be envisaged by observing the streamlines. In the latter flow, which is similar to the observation of the vortex by an observer moving with uniform motion, the streamlines may not identify closed loops at all; however the Galilean invariance of the underlying vortex must be captured by the algorithms.
6. Identifying the center of the decaying vortex. The velocity tends to zero in the center of the vortex while the vorticity attains its peak value. This test may be challenging to assess the performance of the algorithms in low tracer densities.

A wide range of relevant performance tests can thus be designed with piv-image-generator. It is in this sense that we considered this set of flows sufficient for the purpose of testing planar PIV and Optical Flow algorithms. Furthermore, we argue that piv-image-generator is particularly adequate for validation in the sense that it provides a zero-uncertainty ground truth in the form of continuous velocity fields to compare with the results of PIV and OF algorithms.

5. Conclusions

We have presented an image generating software package aimed at creating images that simulate those obtained in a natural laboratory PIV environments, resulting in spots of illuminated tracers against a dark (but normally noisy) background. Image characteristics and error sources are fully parameterized. Images are generated from zero-uncertainty velocity and displacement fields. This configures the key novelty of the piv-image-generator: fully parameterized image generation and zero-uncertainty ground truth is a combination of features never before conjoined in synthetic particle image generators.

Data is exported both for PIV and Optical Flow methods. The piv-image-generator is extensible, since it is open-source, it can be modified by others, and can be easily modified to support additional flow fields. the values of the control parameters can be easily adjusted, using the provided sample scripts.

The piv-image-generator was already used for a comparative study of Optical Flow and PIV methods applied to tracer based images for fluid mechanics velocimetry applications

We strongly believe that piv-image-generator can be used for additional studies for similar applications, even helping in standardizing the source data sets for benchmarking the methods under test. The image generator can also be adapted for other fields of application.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the PhD grant SFRH/BD/137967/2018 from the Portuguese Foundation for Science and Technology (FCT).

This work was partially supported by FCT with the LARSyS UIDB/50009/2020 from the Portuguese Foundation for Science and Technology (FCT).

This research was partially supported by Portuguese and European funds, within the COMPETE 2020 and PORL-FEDER programs, through project RiverCure PTDC/CTA-OHR/29360/2017 from the Portuguese Foundation for Science and Technology (FCT).

References

[1] Adrian R. Twenty years of particle image velocimetry. Springer Verlag; 2005, http://dx.doi.org/10.1007/s00348-005-0991-7.
[2] Particle-imaging techniques for experimental fluid mechanics. Annu Rev Fluid Mech 1992;23:261–304. http://dx.doi.org/10.1146/annurev.fl.23.010191.001401.
[3] Gerontakos P, Lee T. PIV study of flow around unsteady airfoil with dynamic trailing-edge flap deflection. Exp Fluids 2008;45(955). http://dx.doi.org/10.1007/s00348-008-0514-4.
[4] Ma CM, Wang JX, Li QS, Qin H. Vortex-induced vibration performance and suppression mechanism for a Long Suspension Bridge with Wide Twin-Box Girder. J Struct Eng 2018;144(11). http://dx.doi.org/10.1061/(ASCE)ST.1943-541X.0002198.
[5] Blocken B, Druenen T, Toparlar Y, et al. Aerodynamic drag in cycling pelotons: new insights by CFD simulation and wind tunnel testing. J Wind Eng Ind Aerodyn 2018;179:319–37. http://dx.doi.org/10.1016/j.jweia.2018.06.011.
[6] Soid SN, Zainal ZA. Spray and combustion characterization for internal combustion engines using optical measuring techniques – A review. J Energy 2010;36(2). http://dx.doi.org/10.1016/j.energy.2010.11.022.
[7] Ricardo AM, Koll K, Franca MJ, Schleiss AJ, Ferreira ML. The terms of turbulent kinetic energy budget within random arrays of emergent cylinders. J Water Resour Res 2014;50(5):4131–48. http://dx.doi.org/10.1002/2013WR014596.
[8] Berthe A, Kondermann D, Garbe C, Affeld K, Jähne B, Kertzscher U. The Wall-PIV measurement technique for Near Wall Flow Fields in Biofluid mechanics. In: Imaging measurement methods for flow analysis. Springer Berlin Heidelberg; 2009, p. 11–20. http://dx.doi.org/10.1007/978-3-642-01106-1_2.
[9] Chao H, Gu Y, Napolitano M. Survey of optical flow techniques for robotics Navigation Applications. J Intell Robot Syst 2014;73:361–72. http://dx.doi.org/10.1007/s10846-013-9923-6.
[10] Mase K. Recognition of facial expression from optical flow. IEICE Trans Inf Syst 1991;E74-D(10):3474–83.
[11] Garcia F, Cerri P, Broggi A, Escalera A, Armingol J. Data fusion for overtaking vehicle detection based on radar and optical flow. In: 2012 IEEE intelligent vehicles symposium. 2012, p. 494–9. http://dx.doi.org/10.1109/IVS.2012.6232199.

[12] Hongwei W, Zhan H, Jian G, Hongliang X. The Optical Flow Method research of particle image Velocimetry. Procedia Eng 2015;99:918–24. http://dx.doi.org/10.1016/j.proeng.2014.12.622.
[13] Carlier J, Wieneke B. Report 1 on production and diffusion of fluid mechanics images and data. In: Fluid image analysis and description (FLUID) project. Tech. rep., 2005, URL http://fluid.irisa.fr/data-eng.htm.
[14] Schmidt BE, Sutton JA. High-resolution velocimetry from tracer particle fields using a wavelet-based optical flow method. Exp Fluids 2019;60(37). http://dx.doi.org/10.1007/s00348-019-2685-6.
[15] Liu T, Merat A, Makhmalbaf MHM, Fajardo C, Merati P. Comparison between optical flow and crosscorrelation methods for extraction of velocity fields from particle images. Exp Fluids 2015;56(166).
[16] Mendes L, Ferreira RML, Bernardino A. Particle Imaging Velocimetry (PIV) image generator [Source Code]. 2020, http://dx.doi.org/10.24433/CO.1931173.v3.
[17] Batchelor GK. An introduction to fluid dynamics. Cambridge University Press; 2000, p. 286.
[18] Raffell M, Willert C, Scarano F, Kähler C, Wereley S, Kompenhans J. Particle image velocimetry - a practical guide. Springer; 2018, http://dx.doi.org/10.1007/978-3-319-68852-7.
[19] Mendes L, Ricardo A, Bernardino A, Ferreira RML. PIV without cross-correlation: an assessment of optical flow methods. In: 6th IAHR europe congress. 2020.
[20] Mendes L, Ricardo A, Bernardino A, Ferreira RML. Comparison of PIV and Optical flow for river flow applications. River Flow 2020;2020.