IOPscience iopscience.iop.org

Home Search Collections Journals About Contact us My IOPscience

Universal outlier detection for particle image velocimetry (PIV) and particle tracking velocimetry (PTV) data

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2010 Meas. Sci. Technol. 21 057002

(http://iopscience.iop.org/0957-0233/21/5/057002)

The Table of Contents and more related content is available

Download details:
IP Address: 128.95.33.91
The article was downloaded on 30/03/2010 at 03:31

Please note that terms and conditions apply.

Meas. Sci. Technol. **21** (2010) 057002 (5pp) doi:10.1088/0957-0233/21/5/057002

TECHNICAL DESIGN NOTE

Universal outlier detection for particle image velocimetry (PIV) and particle tracking velocimetry (PTV) data

J Duncan[1], D Dabiri[1], J Hove[2] and M Gharib[3]

[1] Department of Aeronautics & Astronautics, University of Washington, Seattle, WA 98195, USA
[2] Molecular & Cellular Physiology, University of Cincinnati College of Medicine, Cincinnati, OH 45267, USA
[3] Graduate Aeronautics Laboratory, California Institute of Technology, Pasadena, CA 91125, USA

Received 30 September 2009, in final form 26 January 2010
Published 26 March 2010
Online at stacks.iop.org/MST/21/057002

Abstract
A generalization of the universal outlier detection method of Westerweel and Scarano (2005 Universal outlier detection for PIV data *Exp. Fluids* **39** 1096–100) has been made, allowing the use of the above algorithm on both gridded (PIV) and non-gridded (PTV) data. The changes include a different definition of neighbors based on Delaunay tessellation, a weighting of neighbor velocities based on the distance from the point in question and an adaptive tolerance to account for the different distances to neighbors. The new algorithm is tested on flows varying from impinging jets to turbulent boundary layers and wakes to wingtip vortices, both PIV and PTV. The residuals for these flows also show universality in their probability density functions, similarly suggesting the use of a single threshold value to identify outliers. Also the new algorithm is found to work with data up to about a 15% spurious vector content.

Keywords: PIV, PTV, outlier detection

1. Introduction

Many techniques have been proposed to deal with the spurious vectors returned from PIV data using variable threshold detection schemes such as in Shinneeb *et al* (2004) and Young *et al* (2004). One of the most currently used methods is the universal outlier detection algorithm of Westerweel and Scarano (2005) which normalizes a residual of measured velocities such that data points in high-gradient areas are not considered outliers simply due to the variability of their neighbors.

While there are many options for the detection of outliers for PIV, PTV has relatively few methods for dealing with these bad vectors. Pun *et al* (2007) used a bootstrapping method to generate sets of data from which statistical probabilities were calculated per vector and used to accurately determine and replace outliers. Unfortunately, this method requires significant computational time due to required iterative processes. Another method, developed by Song *et al* (1999), did not require interpolation of the velocity field or *a priori* knowledge of the flow field, but required that the flow satisfy the continuity equation (assuming incompressible flow). While this method may work in some cases, it can be quite problematic in the case of high Mach number flows and three-dimensional flows.

The approach of the current work is to use the same methodology as Westerweel and Scarano (2005) due to the algorithm's simple nature, easy application, computational efficiency and universality. However, two problems arise when considering the application of a normalized residual test to the scattered data resulting from PTV. Firstly, the issue of identifying neighbors must be solved; in the case of PIV, the gridded data readily lend themselves to the identification of neighbors. Secondly, the data resulting from PTV processing are not equally spaced, and thus, should not have the same influence in determining the viability of a vector in question.

 © 2010 IOP Publishing Ltd Printed in the UK & the USA

Both these problems can be solved with the adoption of a newly defined neighborhood and a weighting of the velocities within the neighborhood by their distance from the point in question.

2. Algorithm

The simple, robust outlier detection technique of Westerweel and Scarano (2005) is based on a threshold value of the normalized residual fluctuation of the velocity of one data point relative to its eight nearest neighbors. This residual also takes into account the minimum possible residual by incorporating a tolerance, possibly corresponding to the precision of the data. This normalized residual is defined in equation (1):

$$r_0^* = \frac{|U_0 - U_m|}{r_m + \varepsilon}, \tag{1}$$

where U_0 is the velocity measured at the data point in question, U_m is the median of its neighbors and r_m is the median of the residual of each neighbor's value, U_i to U_m. A slightly expanded form of equation (1) is shown in equation (2) for clarity:

$$r_0^* = \frac{|U_0 - \mathrm{med}(U_i)|}{\mathrm{med}\,|U_i - \mathrm{med}(U_i)| + \varepsilon}. \tag{2}$$

While this technique works well for the gridded data of PIV, it does not allow for the removal of spurious vectors from the randomly distributed data of PTV. To that end, for the present study, it was determined that the neighborhood must be determined differently than the nearest eight data points. Delaunay tessellation allows a convenient method of defining neighbors as those data points which share triangles (Song *et al* 1999). This results in neighborhoods of five to eight neighbors on average, depending on the spatial arrangement of the data. Similarly, Delaunay tessellation has been used to interpolate vectors from an unstructured grid onto a structured grid (Theunissen *et al* 2007).

Due to the random spacing of dada points, neighbors exist at varying distances from the point in question. This gives rise to the question of weighting the data points. A neighbor that is very far away from the point in question should have less effect as to whether it is deemed an outlier. For this reason, it was decided that all the data points in the neighborhood should be weighted by their distance from the point in question. To be consistent, the velocity at the point in question must also be weighted by a measure of distance, which in this case was taken to be the median distance of the neighbors. To both of these weights, a distance tolerance was added as in the case of Westerweel and Scarano (2005). This tolerance will be discussed below. Since the non-normalized fluctuation was weighted by a distance, the median residual should also be weighted by a distance measure (along with the tolerance mentioned above). The resulting normalized fluctuation is shown in equation (3):

$$r_0^* = \frac{\left|\frac{U_0}{\mathrm{med}(d_i)+\varepsilon_a} - \mathrm{med}\left(\frac{U_i}{d_i+\varepsilon_a}\right)\right|}{\mathrm{med}\left|\frac{U_i}{d_i+\varepsilon_a} - \mathrm{med}\left(\frac{U_i}{d_i+\varepsilon_a}\right)\right| + \varepsilon_a}, \tag{3}$$

where, in this case, ε_a is the adaptive tolerance to be discussed shortly and d_i is the distance from respective neighbors to the point in question. It is now interesting to note what happens if the distance of all the neighbors is the same, d. In this case, the median of the distances is d, and the term $d_i+\varepsilon_a$ can be canceled from all the terms of equation (3) except for the tolerance in the denominator, as shown in equation (4):

$$r_0^* = \frac{|U_0 - \mathrm{med}(U_i)|}{\mathrm{med}|U_i - \mathrm{med}(U_i)| + \varepsilon_a(d + \varepsilon_a)}. \tag{4}$$

It is now clear that if $\varepsilon_a(d + \varepsilon_a) = \varepsilon$, where ε is the tolerance from Westerweel and Scarano (2005), equation (4) becomes identical to equation (2). Knowing that the tolerance has been traditionally set to 0.1 pixel, the new tolerance can be adaptively altered based on a median distance so that $\varepsilon_a(\mathrm{med}(d_i) + \varepsilon_a) = 0.1$. The value of 0.1 (Westerweel and Scarano (2005)) for a tolerance is arbitrary but was found to be a good value.

3. Results

3.1. Universality

In order to test its universality, this proposed technique has been used on a variety of flows, both experimental and synthetic, PIV and PTV. The following table lists the flows and the sources of the data. As can be seen, a wide range of flows has been tested (with the exception of supersonic and microchannel flows), which should provide a good measure of the applicability of this modified normalized residual method.

Testing to compare the results between PIV and PTV, the data from the flow behind a circular cylinder at three different Reynolds numbers (12 000, 18 000 and 24 000) were processed both with PIV (a discrete window shifting method; Westerweel *et al* 1997) and an in-house developed PTV algorithm. Since the current technique can handle gridded and non-gridded data, both resultant PIV and PTV velocity fields were processed with the randomly spaced normalized residual method. Figure 1 shows the fraction of vectors corresponding to a given residual, both normalized and non-normalized. It is clear in the non-normalized case that the PIV and PTV results require different thresholds to properly determine outliers. In addition, there is stratification visible in the different Reynolds numbers of the PIV data, similar to that seen in Westerweel and Scarano (2005). In the normalized case, however, the PIV and PTV data collapse to a single curve, suggesting that a threshold can be universally applied to both the PIV and PTV data. As seen in Westerweel and Scarano (2005), there is no apparent Reynolds number variation in the residual. This shows that the current technique can be applied to both gridded and non-gridded data, using the same threshold with equal efficacy.

Figure 1 (right) shows that portions of the normalized residual data from the PIV evaluation of the turbulent wake data between values of 2 and 3 have a 'bump' profile. We have identified that this is due to the processing of edge data, which only have five neighbors rather than eight, and that the neighborhood bias is not centered about the data point, causing both the residual and the neighborhood fluctuation to be affected similarly for all edge data points.

Table 1. Flow types tested.

Flow description	Source of data
Impinging jet	VSJ Image #301 (Okamoto *et al* 2000)
Surface shear flow	Dabiri (2003)
Turbulent boundary layer	PIV Challenge 03 (Stanislas *et al* 2005)
Wing-tip vortex ($Re = 28\,000$)	Current work
Turbulent cylinder wake ($Re = 12\,000$, 18 000, 24 000)	Current work

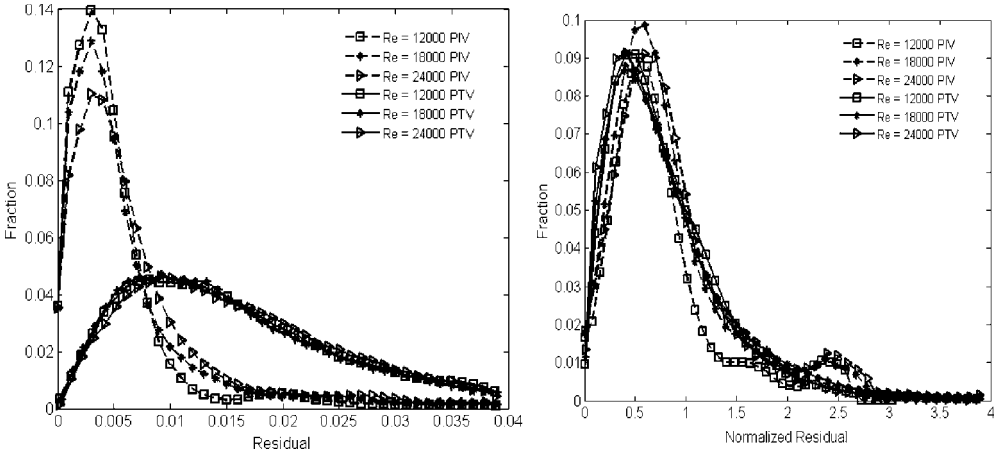

Figure 1. Comparison of the residual (left) and normalized residual (right) of the turbulent cylinder wake flow. The dashed lines indicate PIV data, while the solid lines indicate PTV data.

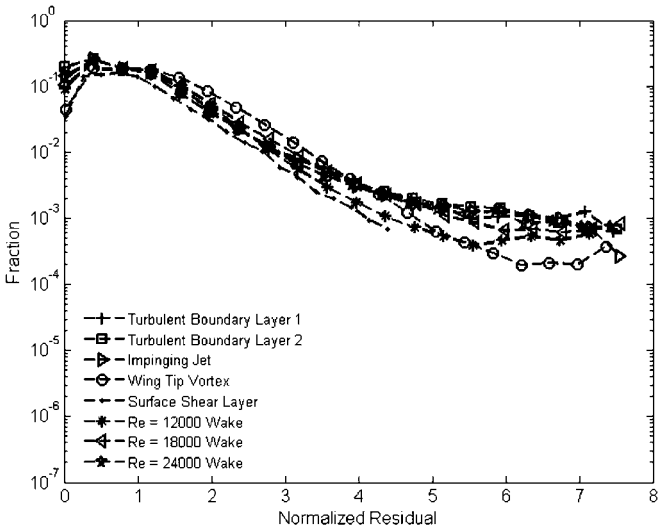

Figure 2. Normalized residual of all the flows tested. Note that the data here are all PTV data and the plot is semi-logarithmic.

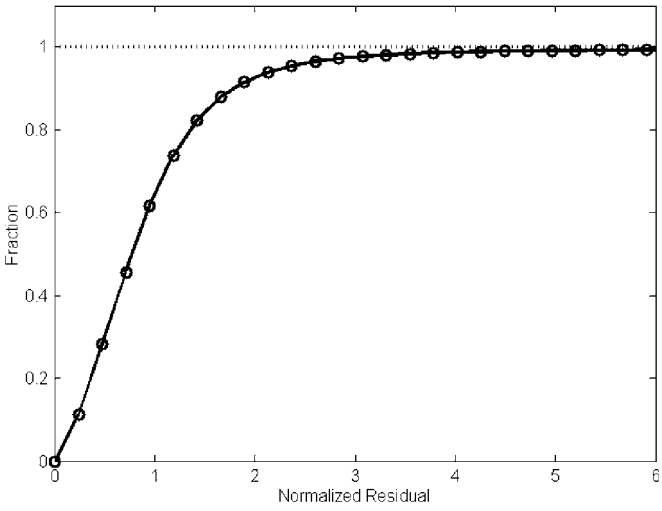

Figure 3. Cumulative density functions for the turbulent cylinder wake ($Re = 18\,000$).

To truly test the universality among different flow types, all the data listed in table 1 were processed with an in-house PTV algorithm and then tested with the current outlier detection algorithm. Figure 2 shows the fraction of vectors (logarithmically) as a function of normalized residuals. As in Westerweel and Scarano (2005), it is found that the data collapses onto a single curve, to which a universal threshold can be applied. In this work, a threshold of 2–4 was found to be applicable depending on the user's confidence in data accuracy. The effect of changing the threshold on the percentage of vectors to be considered outliers can be seen in figure 3, wherein the cumulative density function of the normalized residual is shown for the cylinder wake data. As can be seen from the figure, a threshold of 2 will result in about 8% of the highest normalized residuals to be considered spurious; similarly, a threshold of 4 will result in the 2% highest normalized residuals to be considered spurious.

3.2. Necessity of distance weighting

It is possible to use the same method as mentioned in section 2 to define neighbors while not taking the distance weighting into

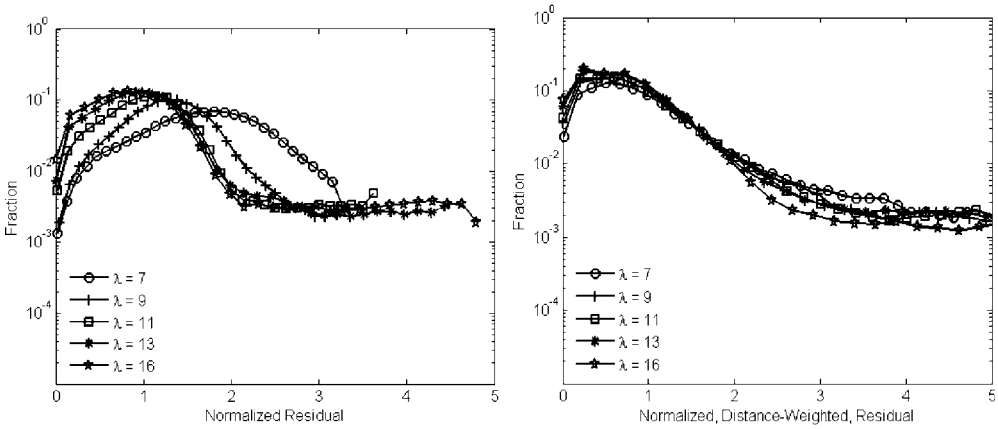

Figure 4. The effect of distance weighting on the normalized residual. The non-weighted residual (left) does not collapse to a single curve, thus not allowing a single threshold, while the weighted residual (right) does collapse.

account. This would essentially weight all neighboring vectors equally and thus not allow one nearby vector to dominate the validation. In practice, this approach generally works well, providing comparable accuracy to the method described above. However, in cases of very small characteristic flow scales relative to mean inter-particle spacing, the ability to pick a universal threshold is lost. To determine this, various synthetic flows of grids of vortices of varying sizes were generated, where the ratio of characteristic flow spatial wavelengths and mean inter-particle spacings (i.e. the ratio between the distance between vortices' centers on a grid and the average distance between particles) were represented by λ. Figure 4 depicts the probability density function versus residual for various values of λ.

It can be clearly seen that the effect of distance weighting allows the use of a single threshold in the determination of spurious vectors. In the unweighted case, higher values of λ seem to be converging, but for very small-scale flows, the probability density functions vary significantly. While an unweighted neighborhood scheme can work in many cases, the weighted algorithm allow for higher spatial gradients while still maintaining universality.

3.3. Outlier detection capability and computational speed

While the universality of the proposed algorithm makes it useful for a variety of flows, the actual necessity is to determine which vectors are spurious. To test the capability of the new algorithm, spurious vectors were added to known clean data (synthetic images, and in the case of PIV, smoothed) from a turbulent boundary layer (Stanislas *et al* 2005). The spurious vectors were added in a random distribution about zero, with lengths up to the maximum length present in the flow. The following table shows the ability of the new algorithm to detect spurious vectors (threshold was set to 2). PIV 1 and PTV 1 correspond to one image pair from the turbulent boundary layer set and PIV 2 and PTV 2 correspond to another pair.

As shown in table 2, it is clear that the algorithm offers decreased reliability when the amount of spurious data increases to around 15%, although the aim should always be to keep the spurious amount below 5%. While the reliability

Table 2. Detection capability on data with added outliers.

Added outliers	PIV 1	PIV 2	PTV 1	PTV 2
0%	0.1%	0.4%	0.9%	0.6%
5%	5.0%	5.3%	5.7%	5.5%
10%	10.0%	10.0%	10.3%	9.9%
15%	14.4%	16.2%	14.2%	13.8%

of the current method seems good even at high spurious levels, caution must be used. When outliers (sometimes several) are adjacent, the results become unreliable due to the increase in neighborhood fluctuation, along with the increase in non-normalized residual. This could result in higher levels of under-detection along with higher levels of over-detection, reducing reliability. For this reason, the current method is not properly suited to highly spurious data.

When there is concern of neighboring spurious vectors, under-detection may occur due to the increase in neighborhood fluctuation and an iterative approach can be used. By iteratively running the outlier detection algorithm, some spurious vectors will be removed on the first pass which will decrease the neighborhood fluctuation for subsequent passes. This approach can be taken until the number of spurious vectors detected does not increase with further iterations. This method will not solve the issue of over-detection, and should only be used when very high spurious rates are expected.

4. Conclusion

A new method of outlier detection for both PTV and PIV data has been developed based on the original algorithm of Westerweel and Scarano (2005). The new algorithm is computationally simple, requiring no interpolation or flow modeling, and its implementation is not dependent upon *a priori* knowledge of the flow. The current method takes two to three times as long as the universal outlier detection method of Westerweel and Scarano (2005), which is mainly due to the time taken by the tessellation process. For example, a 512 × 512 pixel image was processed with 50% overlap 32 × 32 pixel interrogation windows; the original universal outlier

method took 0.3 s, while the current method required 0.85 s (the processing was done on a Dell 8300 with a 2.6 GHz Pentium 4). It was tested on several different flow cases, ranging from an artificial impinging jet, to turbulent boundary layers and wakes, to wing-tip vortices, both PIV and PTV. In order to determine the neighborhood around a given data point, Delaunay tessellation was used. The distances from individual neighbors to the point in question were used as weights to the velocities of the neighbors to account for the differing inter-particle spacing. This method works equally well for PIV and PTV up to a level of spurious data of about 15%, far higher than should be encountered with good experimental techniques. Beyond the 15% level, adjacent outliers affect one another and are either not detected or cause correct vectors to be misidentified as outliers. The method involves an adaptive tolerance, which can be related to the tolerance of 0.1 pixel given by Westerweel and Scarano (2005). A threshold of 2–4 for the normalized residual was found to work well for all flows tested, although it is recommended that a threshold of 2 be used for any flow in which a large percentage of spurious vectors are anticipated.

Acknowledgments

The authors gratefully acknowledge the support of the National Institutes of Health (R01 RR023190-04) to DD, JRH and MG. The authors also gratefully acknowledge the insightful discussions and suggestions of Professor Jerry Westerweel who has added much to the value of this manuscript. We also thank Namiko Saito for generously sharing her wing-tip vortex flow data for this study, as well as helping to acquire the turbulent cylinder wake data.

References

Dabiri D 2003 On the interaction of a vertical shear layer with a free surface *J. Fluid Mech.* **480** 217–32

Okamoto K, Nishio S, Saga T and Kobayashi T 2000 Standard images for particle-image velocimetry *Meas. Sci. Technol.* **11** 685–91

Pun C S, Susanto A and Dabiri D 2007 Mode-ratio bootstrapping method for PIV outlier correction *Meas. Sci. Technol.* **18** 3511–22

Shinneeb A M, Bugg J D and Balachandar R 2004 Variable threshold outlier identification in PIV data *Meas. Sci. Technol.* **15** 1722–32

Song X, Yamamoto F, Iguchi M and Murai Y 1999 A new tracking algorithm and removal of spurious vectors using Delaunay tessellation *Exp. Fluids* **26** 371–80

Stanislas M, Okamoto K, Kähler C and Westerweel K 2005 Main results of the second international PIV challenge *Exp. Fluids* **39** 170–91

Theunissen R, Scarano F and Riethmuller M L 2007 An adaptive sampling and windowing interrogation method in PIV *Meas. Sci. Technol.* **18** 275–87

Westerweel J, Dabiri D and Gharib M 1997 The effect of a discrete window offset on the accuracy of cross correlation analysis of digital PIV recording *Exp. Fluids* **23** 20–8

Westerweel J and Scarano F 2005 Universal outlier detection for PIV data *Exp. Fluids* **39** 1096–100

Young C N, Johnson D A and Weckman E J 2004 A model-based validation framework for PIV and PTV *Exp. Fluids* **36** 23–35