Development of a Vision-Based Particle Tracking Velocimetry Method and Post-Processing of Scattered Velocity Data

Micah Philip Paul

A thesis submitted in partial fulfillment of the requirements for the degree of

Masters of Science in Aeronautics and Astronautics

University of Washington
2012

Committee:
Dana Dabiri
Antonino Ferrante

Program Authorized to Offer Degree: Department of Aeronautics and Astronautics

University of Washington

Development of a Vision-Based Particle Tracking Velocimetry Method and Post-Processing of Scattered Velocity Data

Micah P Paul

Chair of the Supervisory Committee:
Professor Dana Dabiri
Department of Aeronautics and Astronautics

In this thesis, a new vision-based hybrid particle tracking velocimetry (VB-PTV) technique is described and methods of processing randomly scattered velocity data investigated. The VB-PTV technique uses a feature matching method from computer vision theory which relies on the principles of proximity, similarity, and exclusion, meaning that it seeks to match one feature to one feature in subsequent images, and it favors matches which are close to one another and "look" similar. By constructing a matrix which takes these principles into account and performing singular value decomposition, a straightforward method of matching is developed which can give accurate matching results in a wide variety of flows. PIV velocity information is used to provide guidance to the matching algorithm. In addition, matches are made iteratively and validated by an outlier detection scheme. When this method is tested on synthetic images it results in matches which are typically reliable more than 98% of the time. A simple modification to the principle of proximity is introduced which reduces the PTV method's errors in highly shearing flow, as well as improving performance in general for various flow types.

Finally, a natural neighbor-based interpolation technique is investigated for use in estimating flow derivatives using scattered velocity data. This interpolation method is compared with other existing techniques in terms of accuracy, sensitivity to noise, computational efficiency, and spatial resolution. It is found that the natural neighbor interpolation is less accurate than RBF and kriging interpolation methods, and more sensitive to noise, despite the use of a denoising technique.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank those who made this research possible. My thanks go to the University of Washington staff and faculty who accepted me into this program and provided me with the necessary education. Special thanks go to Professor Dana Dabiri, who provided the guidance and motivation to finish this work, and to the National Institute of Health (R01RR023190-04), National Science Foundation (0929864), and the United States Air Force for providing the necessary financial support for this project. I would like to thank Joey Duncan and Ying-Chen Casper Lei, who worked on this project before me and on whose work so much of this thesis is based. Finally, though not least importantly, my sincere gratitude goes to Wei-Hsin Tien, whose help and suggestions during the last two years, as the senior student on this project, have been indispensible.

Disclaimer: The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Chapter 1. INTRODUCTION

1.1 PARTICLE IMAGE AND TRACKING VELOCIMETRY

In the past several decades, particle velocimetry has been a widely used tool in examining the behavior of fluid flows globally, rather than at discrete points in time, such as measurements recorded using pitot tubes or hot wire anemometers. Within the field of particle imaging, two separate methods have been developed and gained widespread use: particle image velocimetry (PIV) and particle tracking velocimetry (PTV). PIV uses cross-correlation of segments of an image to give an estimate of the average velocity in a local region, while PTV attempts to identify and match individual particles between image frames. Early techniques relied on streak photography and manual measurements of particle locations and velocities, as described by such pioneers in the field as Agui and Jimenez in 1987. In the 1990's the power and availability of high-speed computers and CCD cameras allowed much of the particle imaging process to be automated, removing the necessity for manual measurements on film prints. In 1996, Baek and Lee advanced PTV significantly by introducing a two-frame matching algorithm to replace the less accurate and more time intensive four-frame methods that had been used previously. Possible matches were assigned values which described their "match probability" and their "no-match probability" based partly on heuristic parameters such as the maximum expected velocity in the interrogation area. Though this was an improvement compared with nearest-neighbor approaches, it was limited to low gradient flows, in part due to a quasi-rigidity assumption in local areas of the flow.

At nearly the same time, PIV and PTV techniques were melded when PIV results were used to guide particle matching in a "super-resolution PIV" technique (Keane *et al*, 1995). The additional matched particles were envisioned as a means of increasing the resolution of PIV, and

with the addition of a Kalman filter and χ^2 testing, the method was made more robust and accurate (Takehara *et al,* 2000). Alternatively, Cowen and Monismith (1997) used results from PIV performed on a pair of singly pulsed images to guide a particle tracking algorithm. By interpolating the PIV results a small (3 x 3 pixel) interrogation window was generated in the second image, and if a single particle was identified within this window it was considered a match to the particle of interest in the first image frame. More recently this idea of PIV-guided PTV was combined with Baek and Lee their concept of match probability (Kim and Lee, 2002). Instead of using global heuristic parameters, the matching algorithm was guided by more local velocity data from PIV. This simplified detection of erroneous matches and improved the number and accuracy of matches that could be made compared with the original technique.

Other methods which have been developed in the past several years include the deterministic annealing technique (Stellmacher and Obermayer, 2000) which constructs and attempts to minimize a cost function based on the goodness of match of a group of particles in each image frame. Song *et al*. used Delaunay tessellation to match triangles, rather than individual particles, in a pair of images (2002). More recent PTV methods have used a variational approach to matching, as in Ruhnau *et al,* (2005). This technique, somewhat like the deterministic annealing method, examines a group of particle images, in this case the entire image, and makes matches which satisfy a minimization problem which includes local flow information as well as a global smoothness constraint.

Uemura *et al*. (1989) had developed a method of tracking which was similar to PIV in that it cross-correlated a small region (35 x 35 pixels) around each particle with regions around candidate particles to identify matches. This method, called Enhanced PTV (EPTV), was updated by Mikheev and Zubtsov (2008) by including information about particle image sizes. A

method developed by Ponchaut and Mouton (2005) uses a technique similar to that developed by Kim and Lee (2002) which guides a matching algorithm with PIV and initial PTV results. The iterative matching process combines the results of a correlation method in high density regions with a simple tracking method in low density regions to construct a hybrid velocity field which guides the final particle matching routine. Very recently even more algorithms have been proposed by Brevis *et al*. (2011), who combined the correlation methods of, for example, Uemura (1989) and Ruhnau (2005) with the relaxation methods of matching which use match probabilities (Baek and Lee, 2006). By combining the complementary methods, the new algorithm was useful in regions of high flow gradients as well as low and high particle image density regions and enjoyed overall better performance than the component techniques. Panday *et al*. (2011) developed a technique for matching features in stereo images which was used in an ant colony optimization study, but which could be applied to general particle tracking as well. Schindler *et al*. (2010) modified a polar coordinate system similarity method based on nearest-neighbor cluster matching which performed very well even in flow fields with great variance in seeding density.

PIV algorithms apart from particle tracking also have a well-developed history of improvements and innovations. Because PIV gives an average displacement of particles within a region, it is less sensitive to noise and can give measurements with very low errors when compared to PTV methods. But because of their nature, PIV methods must always strive for greater spatial resolution. Nogueira *et al*. (2005) demonstrated that PIV could resolve wavelengths as small as twice the grid node spacing, though errors were as large as 100%, while wavelengths of 4 to 8 times the node spacing could see errors closer to 10%. In addition to spatial resolution, PIV methods, because they correlate large sections of an image, give an

average displacement and therefore underestimate velocity gradients (Scarano, 2003). Because PTV methods examine individual particles, they will not suffer from this averaging effect even in strong gradients. Similarly, PTV can achieve higher spatial resolution, limited by the mean particle spacing ($1/\sqrt{\pi p}$, where p is the particle image density, or number of particle images per image area in pixels) rather than interrogation window and window overlap, as in PIV.

1.2 POST-PROCESSING OF PTV DATA

Because PIV data exists on a regularly spaced grid, extracting flow information can be as simple as applying finite difference methods (see, for example, Raffel *et al*., 1998). PTV data is randomly scattered and so some method must be used to fit the data, interpolate the data onto a regular grid, or otherwise derive useful flow characteristics like shear and vorticity. Within the field of particle tracking velocimetry, one of the most common methods of interpolation has been interpolation of scattered velocity data onto a grid (see, for example, Agui and Jimenez, 1987; Spedding and Rignot, 1993). These have the advantage of being simple to implement, and because each interpolation node is influenced by a number of PTV vectors, they tend to have a smoothing effect on noise in the field. However, this also means that the spatial resolution gained by PTV over PIV is forfeited when shear or vorticity is calculated. Since velocity measurements are not usually the end goal of particle velocimetry, it is desirable to obtain flow parameters such as shear stress (Dong and Meng, 2001), vorticity (Luff *et al*., 1999; Fouras and Soria, 1998; Foucaut and Stanislas, 2002), or dissipation rate (Saarenrinne and Piirto, 2000; Tanaka and Eaton, 2007) with as much accuracy and spatial resolution as possible. An alternative to grid-based interpolation is to find an analytic fit to velocity data (Abrahamson and Lonnes, 1995; Fouras and Soria, 1998) and determine flow gradients from this fitted surface.

1.3 OBJECTIVES

Though numerous PIV and PTV techniques currently exist, there are weaknesses in both that leave room for improvement. As mentioned earlier, PIV, because it is a statistical average of local particle displacements, can give accurate velocity data, but with a limited spatial resolution. PTV can in theory achieve a higher spatial resolution because it examines individual particles, but due to difficulties in identifying overlapped particle images, PTV has traditionally been used to process images with relatively few particles.

To address these shortcomings, a new vision-based PTV (VB-PTV) technique is developed for use at the University of Washington. A Cascade Correlation Method is used to identify overlapped particle images (Angarita-Jaimes *et al*., 2009), allowing for accurate particle centroid estimates and resulting in a denser velocity data field. A feature matching technique developed by Scott and Longuet-Higgins (1991) is used to match particle images in sequential image frames. This matching method is improved by embedding it within an iterative process that makes use of a newly developed outlier detection scheme (Duncan *et al*., 2010) to verify matches. These algorithms are described further in chapter 2 and their implementation within the VB-PTV technique is described in chapter 3. Experimental results are presented in chapter 4. A simple modification to the Scott and Longuet-Higgins technique is proposed in chapter 5 and updated experimental results reported. Strain rate is estimated by interpolating simulated PTV data using a Laplace interpolation method and compared with existing interpolation techniques in chapter 6.

Chapter 2. PTV ALGORITHMS

2.1 VISION-BASED PTV

In the field of computer vision, the ability to match features from one image frame to another in order to discern motion is of great importance. When features are large and distinct, matching them across multiple frames can be simple. However when features are small and indistinct, as in the case of small particles, matching becomes more problematic. Two principles have been developed pursuant to solving this issue which are based on the study of human vision: the "Principle of Exclusion" and the "Principle of Proximity" (Scott and Longuet-Higgins, 1991). The first principle requires that one feature in one image is not associated with multiple features in another image. In terms of particle motion, this would mean a one-to-one correspondence between particles in each image. The proximity principle simply states that features are more likely to be matched the closer they are to one another.

To include the proximity principle, Scott and Longuet-Higgins first constructed a "proximity matrix" *G* based on Gaussian weighted distances between all features in images *I* and *J* which is given by

(1)

Hereafter, features (particles) in the first image frame (image *I*) will be referred to as *target* features or particles, and all the possible matches which exist in the second image frame (image *J*) will be referred to as *candidate* features. In the above matrix, r_{ij} is the distance between a target feature *i* and a candidate feature *j*, and σ is a characteristic distance. The proximity matrix will contain elements ranging from unity, if there is zero distance between features *i* and *j*, to zero when two features are infinitely distant from one another. This proximity matrix is then

used to construct a "pairing matrix" *P* which is an orthogonal matrix which will maximize the inner product *P:G*. This is accomplished by performing singular value decomposition on the *G* matrix,

(2)

T and *U* are orthogonal matrices, and *D* is a non-negative diagonal matrix with the same dimensions as *G*. The matrix *D* is replaced with a rectangular identity matrix of the same dimensions and used to find the pairing matrix

(3)

This pairing matrix is used to directly find matched features. It has dimensions $m \times n$, where *m* is the number of target features, and *n* the number of candidate features. It can be shown to maximize the inner product with *G*, and it has rows which are mutually orthogonal. Those rows in the pairing matrix index target features in the first frame, while the columns index the candidate features in the second frame; the larger an element P_{ij} is, the greater the correspondence between features *i* and *j*. If the element P_{ij} is the greatest value in the i^{th} row and j^{th} column, then we consider those features to be a match. The mutually orthogonal rows of the P matrix tend to disallow a strong correspondence between one target feature with more than one candidate feature, or vice versa. In this way the exclusion principle is satisfied without explicitly enforcing it, as has been done in other similar vision methods (Ullman 1979). For further mathematical details the reader is directed to Scott and Longuet-Higgins (1991) as well as Schonemann (1966).

The computer vision technique described above can clearly be applied to particle tracking. In theory, its implementation is also fairly straightforward, requiring only singular value decomposition, matrix multiplication, and a maximum value search. There are, however,

some obstacles to this technique's implementation as a particle matching method. Scott and Longuet-Higgins found that the addition of "rogue points," or features which exist in one image frame but not the other, could corrupt their results. Such rogue points will almost invariably appear in experimental PTV images as seeding particles travel in and out of the interrogation area or through an illuminating laser sheet, or simply due to optical phenomena which cause the erroneous identification of particles which may not actually exist. The impact of rogue points on experimental results is discussed in section 4.7.

It was also found that this algorithm was unable to correctly match features across a large rotation, which could clearly be problematic in any study of vortical flows. Luo and Hancock (2002) found that the critical angle of rotation was near 20 degrees, which is larger than what should be found in a typical PTV dataset if the experiment is performed properly.

Finally, the Scott and Longuet-Higgins technique requires selection of a characteristic distance, σ. In the original paper, one value of σ was selected for an entire image set. Scott and Longuet-Higgins (1991) suggested that this value might relate to the average displacement of all features between image frames, and in experiments Pilu (1997) found that σ should approximately match the actual displacements found in the images. In a PTV application a great range of displacements could exist in one image frame, and so an adaptive σ is desirable; the natural choice is a displacement estimated from PIV results.

Others in the field of computer vision expanded on Scott's and Longuet-Higgins' work by including another principle common in the field: the "Principle of Similarity," (Pilu, 1997) which unsurprisingly prefers matches which "look" similar to one another. This was accomplished by modifying the construction of the proximity matrix G by including a normalized cross-correlation coefficient, C_{ij}, which is described as

(4)

This term is generated by a cross-correlation of pixel intensities within a window of dimensions $W \times W$ centered on the target feature i and similarly sized windows centered on all candidate features j. and are the mean pixel intensities within each correlation window, and and are the standard deviations (not summations; a capital sigma is used to avoid confusion with the characteristic distance) of those pixel intensities. The value of C_{ij} can range from -1 for a completely uncorrelated candidate feature to 1 for an identical feature. This can be incorporated into the proximity matrix by various means, one of which is shown below:

(5)

In this formulation, the similarity coefficient is Gaussian weighted, where γ is a correlation parameter which controls the rate of decay of similarity weighting. This value is set to 0.4, per Pilu (1997). This modification reduces the impact of rogue features and produces more valid matches than the original algorithm.

The particle tracking algorithm developed here combines the feature association algorithm proposed by Scott and Longuet-Higgins (1991) with the correlation-based "similarity" term recommended by Pilu (1997), as well as PIV results which provide an estimate for the proper characteristic distance, σ, and an outlier detection method (Duncan *et al*., 2010) which is embedded within an iterative matching process. This method is dubbed vision-based PTV, or VB-PTV. Further details are provided in chapter 3.

2.2 PARTICLE LOCATION IDENTIFICATION

A particle matching method can be considered reliable if the matches made between features in a series of images are correct. The accuracy of a PTV method depends on the ability to accurately identify the location of particle images based on the pixel intensity information. Success in this step is foundational to a useful PTV method, and every effort should be made to achieve it. In order to take advantage of PTV methods' high spatial resolution, a high seeding density is desirable. This, however, leads to particle image overlap, which can cause multiple overlapped particles to be identified as one (Ponchaut, 2005), and can greatly increase errors in particle location estimates (Marxen *et al*., 2000).

Previous work has shown that a particle mask, or a Gaussian "model image" of a particle, can be cross-correlated with overlapped particles to estimate the particle center locations (Stellmacher and Obermayer, 2000; Takehara and Etoh, 1999; Saga *et al*., 2003). Angarita-Jaimes *et al*.(2009) suggested that this method could be improved by repeating the cross-correlation process on the correlation surface resulting from the first pass. This narrowed the correlation peak and decreased the critical distance of separation between overlapped features where two particles could no longer be identified. The method used in the current work is an extension of this cascade correlation method (CCM). The correlation of correlation surfaces is repeated with a smaller and smaller Gaussian mask in order to narrow the correlation peaks as much as possible. This results in a number of well defined peaks. In the CCM and other similar methods, fitting of the correlation peak can be biased towards integer values due to the discretization of digital images. Therefore, the modified CCM method uses the number and locations of correlation peaks to perform a least squares fit of the cluster of particles being examined. That is, if the modified CCM method identifies 4 peaks, 4 Gaussian particle models

will be fitted to the pixel intensity surface by varying their location, radius, and peak value. This method takes advantage of the CCM's ability to separate overlapped particles, but avoids the errors introduced by repeated cross-correlation. Figure 1 shows the results, found in Lei, *et al.* (2012), of using this modified cross correlation method in the presence of Poisson and Gaussian noise. In Figure 1a, random Poisson noise is added to overlapped particle images, and the error in their locations is plotted against the amount they are overlapped as a percentage of their diameters. The performance of the modified CCM method is compared with the original CCM method (Angarita-Jaimes *et al.*, 2009) using signal to noise ratios of 2.5, 10 and infinity (no noise). Even at the lowest SNR, errors are below 0.2 pixels for overlaps as large as 50% of the diameter. Figure 1b shows similar results after applying 2.5% and 5% Gaussian noise to the particle images. Errors remain below 0.09 pixels for overlaps of 40% or below, and remain below 0.25 pixels for 50% overlap with 5% Gaussian noise added. Essentially, these results show that the modified CCM method can detect particles overlapped by as much as 50%, meaning that PTV can be applied to images with a higher particle image density. A much more detailed description of this method which includes equations, a flow chart of the process, and results of experiments with overlapped particle images, can be found in Lei, *et al*. (2012).

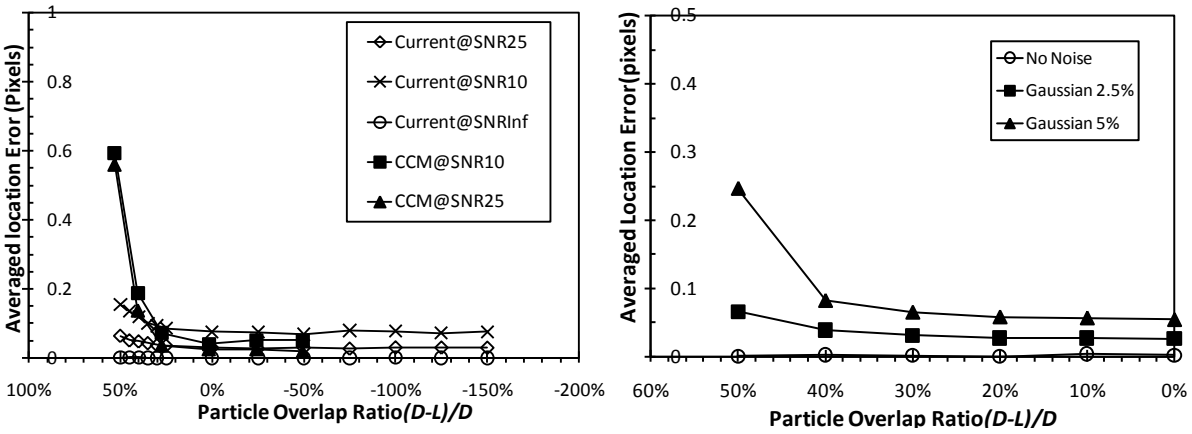

Figure 1. Averaged particle location error versus particle overlap ratio between two particles for (a) different Poisson noise levels (b) different Gaussian noise levels

Chapter 3. ALGORITHM IMPLEMENTATION

3.1 MODIFIED VISION MATCHING

Once particle locations are known, the matching algorithm can be implemented. The Scott and Longuet-Higgins (1991) technique is used, as described in chapter 2. In order to choose a characteristic distance, σ, PIV is performed to obtain an initial estimate of local displacements in the flow. The PIV displacement data, which exists on a regularly spaced grid, is interpolated to individual particle locations, and the characteristic distance is typically set to be twice this interpolated value. Scott and Longuet-Higgins (1991) found that an overestimate of the actual displacement was preferable to underestimates, and this was supported by experiments with the current VB-PTV algorithm (see Scott and Longuet-Higgins (1991) and chapter 5).

Performing the matching process described earlier results in a pairing matrix, $P_{ij,}$ whose elements can be used to determine matches. In order to make the matching algorithm more robust, the process is made iterative in two ways. First, an outer iteration loop (termed the particle removal loop) finds and removes matched particles from the "to-match list" (that is, particles which have yet to be matched). Not every particle will be matched after one pass, or not

all matches will be allowed because they are outliers. By removing those particles which are most easily matched and constructing a new pairing matrix, matching the remaining particles becomes simpler since there are fewer candidate particles to pair with any given target particle. The second iterative loop is nested within the particle removal loop just described and is termed the validation loop. After matches are found, these results are submitted to an outlier detection scheme dubbed the modified universal outlier detection method (Duncan *et al*., 2010). If a match is considered an outlier, that element in the pairing matrix is set to zero and a different element will become the maximum value for that row and column. When the pairing matrix ceases to change, this loop ends and the particle removal loop continues. The outer particle removal loop ends when no more matches are made and validated. A flow chart of this process is shown in Figure 2.

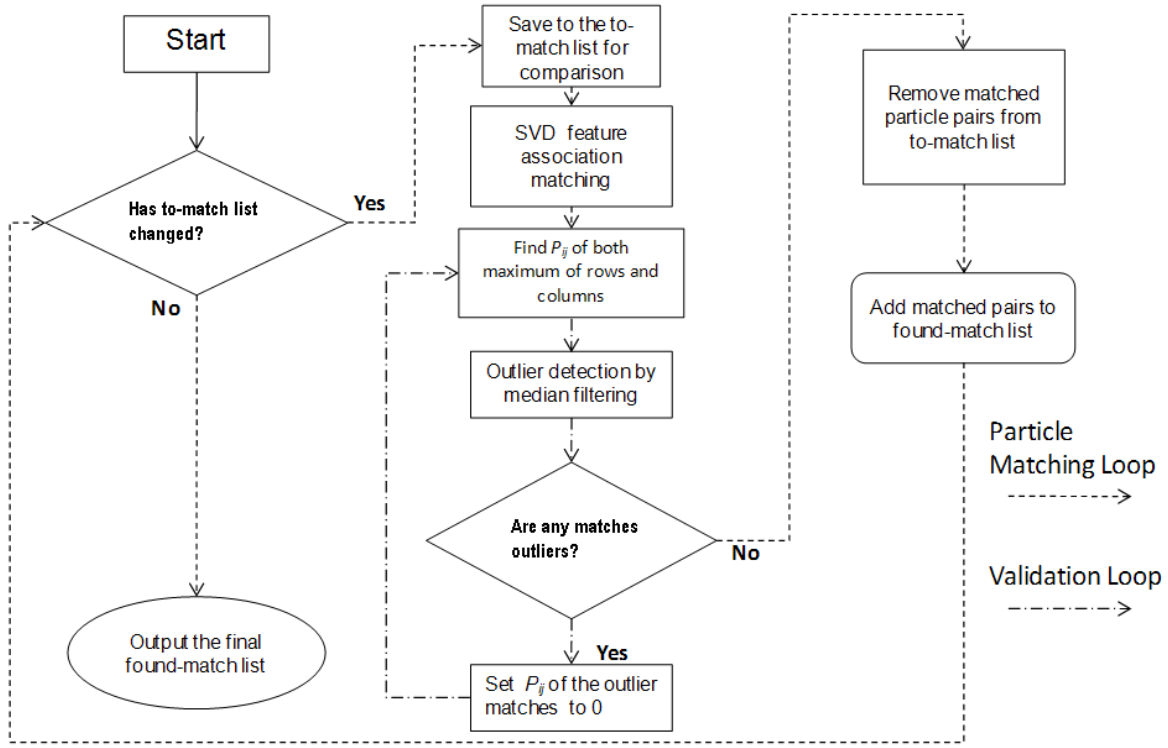

Figure 2. Matching algorithm flow chart.

3.2 COMPUTATION TIME

An experimental image may contain thousands of particle images, and the effort of locating each one, performing a matching method, validating matches, and any other necessary steps can become computationally expensive. In the interest of simplifying the matching process and so reducing the computation time, images are broken into smaller overlapping sub-windows and particle matching performed on each separately. It was found that doing this greatly reduced the computational effort while maintaining accuracy. A Dell Precision PWS490 Intel® Xeon® CPU E5345 @2.33GHz with 16.00GB of RAM was used to process a 512 x 512 image using various sizes of sub-window. The number of matches made in each case is recorded in Table 1, where it can be seen that using smaller windows has little to no effect on the number of matches being found. Similarly, images with different particle image densities (defined as the number of particles per image area in square pixels) were processed using different sub-window sizes and the computation time is recorded in Figure 3. At the highest density (0.06), the computation time is reduced from 14 hours to 14 minutes, and as seen in Table 1, this came at the cost of only 12 fewer matches out of more than 15000.

Table 1: Effect on PTV results from breaking 512x512 images with different particle image densities into overlapping windows

Window Size	**64x64**	**128x128**	**256x256**	**512x512**	**Total No. of Particles**
Matches (0.01 density)	2592	2592	2592	2592	2611
Matches (0.03)	7828	7828	7829	7829	7879
Matches (0.06)	15384	15395	15397	15396	15512

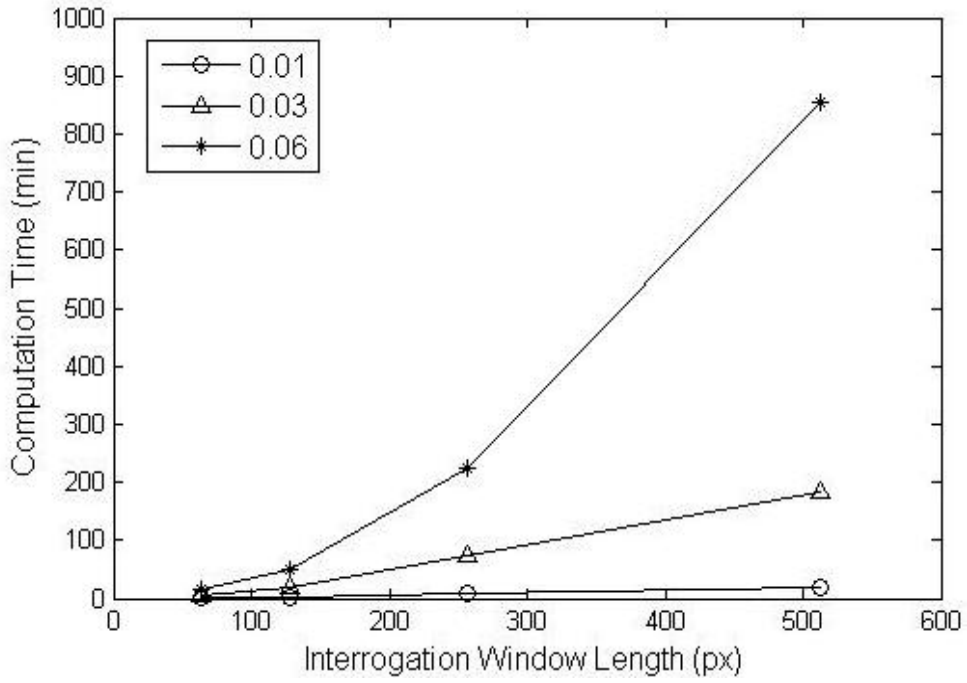

Figure 3. Computation time as a function of sub-window size and particle image density (0.01 – 0.06).

These losses are caused by the matching method's sensitivity to unmatchable particles which cross the edges of an image or sub-window. To minimize this source of error, it is recommended that the overlapped region between windows be at least twice as large as the expected displacement of particles (e.g. if a 10 pixel displacement is expected, 64 x 64 windows with 50% overlap will have an overlapped region of 32 pixels, which is sufficient to minimize this source of error). Results recorded in this paper typically come from PTV runs using 64 x 64 or 128 x 128 sub-windows with 50% overlap.

Chapter 4. EXPERIMENTAL RESULTS

4.1 GENERATION OF SYNTHETIC IMAGES

The particle matching algorithm is now tested using synthetic images where the analytic flow and particle locations are known. Synthetic particle images are generated using four

parameters: peak intensity I_o, representative radius, d_r, and peak location (x_c,y_c).Their shape is Gaussian (Raffel *et al*., 1998), typically described by

(6)

A top hat light sheet intensity profile is assumed in most images; that is, I_o is constant throughout a synthetic image. Peak locations are randomly generated in the first image and then the set of peak locations in the second image is found using the analytic flow field. Images are generated by summing the intensities of all particle images which exist at each pixel. Synthetic flows are 2D and so it is assumed that there is no intensity variation between the first and second image.

The number of particles per image area, or particle image density, varies from 0.01 to 0.06. At high particle image densities, it is possible that the large number of particles could interact with and change the flow characteristics. In a typical particle velocimetry experiment with 0.06 particle image density, a laser sheet thickness of 1mm, tracer particle diameters of 10 μm, and an image area of 10000 mm^2, the volume fraction, Φ_p, of particles is $2.09x10^{-7}$. Elghobashi (1994) shows that the effect on turbulence at these conditions is negligible, and so there are no two phase flow effects. Therefore the synthetic images used in the following experiment are acceptable representations of an actual flow, even with high particle image densities.

4.2 VB-PTV Results from a Moving Wall Flow (Stokes' First Problem)

The matching algorithm is tested using a moving wall flow (Stokes' first problem). Images dimensions are 512 x 512 pixels. The VB-PTV method is guided with results from a correlation method (PIV) which uses window-shifting and multi-passes which typically results in displacement data on a uniform grid with nodes separated by 8 pixels.

Two measures of performance are adopted from Ruhnau (2005): the match yield and reliability, given as percentages. Match yield (E_Y) is defined as the number of correct matches made (n) over the total number of possible matches (v). Reliability (E_R) is defined as n divided by the total number of matches made (d). That is, match yield gives a measure of the algorithm's ability to make a correct match when presented with a pair of particle locations, and reliability gives a measure of the accuracy of those matches. These measures of performance can be expressed in the following way:

$$-\qquad (7)$$

$$-\qquad (8)$$

A match is considered "correct" if it falls within a 1 pixel tolerance (EPTV: Mikheev and Zubtsov (2008), NRX: Ohmi and Li (2000), VAR: Ruhnau *et al* (2005)). A tighter 0.5 pixel tolerance is also included to further quantify the performance of the VB-PTV algorithm. Particle yield is changed subtly from the definition above and defined as the number of matches made over the number of particles within the image. Finally, the root mean square (RMS) error recorded in the following results measures the error between the found particle displacements and the true displacements particles should experience, as given by the analytic flow field.

Synthetic images are generated using the analytic flow field from Stokes' first problem with a velocity profile described as:

$$—\qquad (9)$$

Here, U=10, μ = 5, and t=200. Parameter t determines the sharpness of the velocity gradient Particle image density is 0.03 and particle image diameters are 4 pixels.

Results from two experiments are shown in Table 2. The first experiment used the exact known locations of particles in the matching process. That is, the particle location identification step described in section 2.2 was skipped. In the second experiment particle locations were identified and then these coordinates fed into the matching algorithm.

It can be seen that the matching algorithm can achieve great accuracy when given accurate particle location information, even achieving errors as low as machine precision (10^{-16} in Matlab®). When particle locations are extracted with the particle location algorithm, errors are introduced and the RMSE value rises to 0.12 pixels while the reliability of matches drops to 99.6% and 98.9% for 1 px and 0.5 px tolerances, respectively. A scatter plot of the VB-PTV results from the moving wall flow is shown in Figure 4. The majority of data points have negligible error, and so they appear as a solid black line. The analytic velocity profile is superimposed on this scattered data as a dashed white line. Additionally, the scattered PTV data is fitted to the known solution of a moving wall flow, where parameters U, μ, and t are allowed to vary to minimize the sum of the squared error between the fit and the scattered data. The resulting parameters from this curve fit are U=9.99, μ=5.06, and t=196.91. The R^2 value for the fitted curve is 1.000 and the average error of the three parameters is 0.93% when compared with the exact parameters of U=10, μ = 5, and t=200. Naturally, the fitted curve is very accurate in part because the exact solution is known and used as a guide to the curve fitting problem, but this is meant to give one more quantifiable measure of the accuracy of the matching method.

Table 2: PTV results using exact particle locations and results from particle finding algorithm. Particle image density 0.03

	Particles Found	Matches Found	Match Yield with 1 Px Tolerance	Reliability with 1 Px Tolerance	Reliability with 0.5 Tolerance	RMSE (pixels)
Known particle locations	-	3777	99.8%	100%	100%	10^{-16}
Unknown locations	2865	2820	98.1%	99.6%	98.9%	0.119

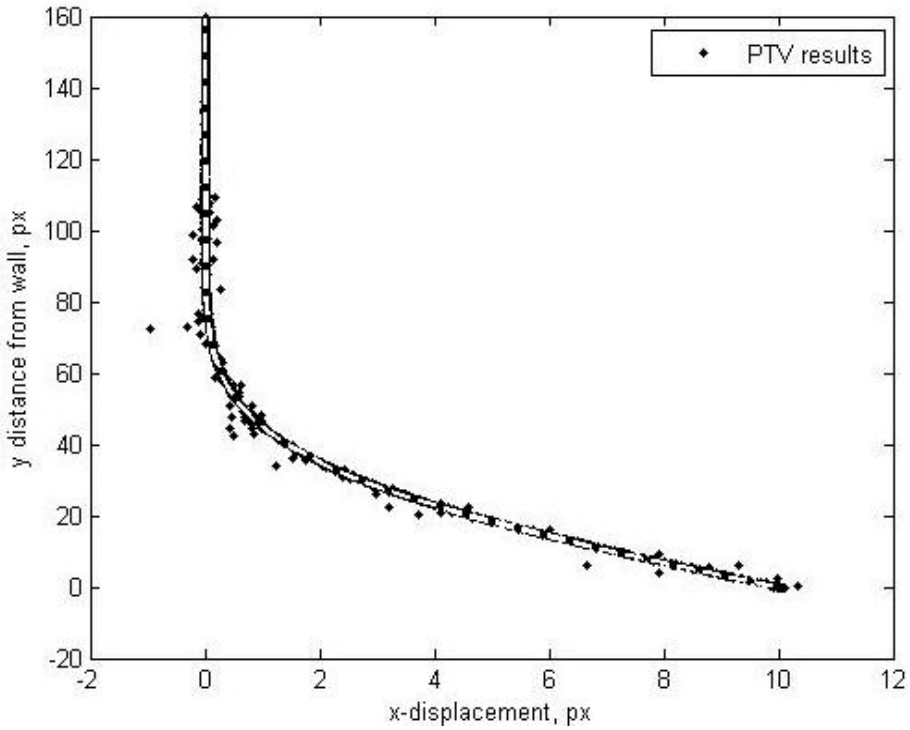

Figure 4. PTV velocity profile for moving wall flow compared with analytic flow profile which is shown as a dashed white line superimposed on the scattered PTV data.

4.3 Errors Due to Gradients, Large Displacements, and PIV

A number of experiments are performed to describe the current matching algorithm's performance in flows with high gradients, large displacements, and the impact of PIV guidance on the VB-PTV results. As mentioned earlier, PIV methods underestimate flow gradients (Scarano 2003) because they correlate regions of the flow in which particles with larger

displacements will leave the interrogation window at a higher rate than those with smaller displacements. A generic PTV method will not suffer from this biasing since it examines individual particles. The current method is a hybrid of PIV and PTV, using correlation results to estimate a local characteristic displacement, therefore we wish to learn whether the advantages of PTV are lost because of the weaknesses of PIV. To isolate the response of the VB-PTV algorithm to gradients, a new synthetic flow is introduced: simple 1D uniform shearing flow. These shearing regions are "capped" once the flow displacement reaches a maximum value, and since it was found that the performance of the VB-PTV algorithm was affected by the maximum displacements in the gradient region, two cases are presented; one flow which shears until the displacement reaches ±7 pixels and one which is capped at ±25 pixel displacements. The particle image density is 0.01. These velocity profiles are shown in Figure 5 for flow with a gradient of 0.5 px/px.

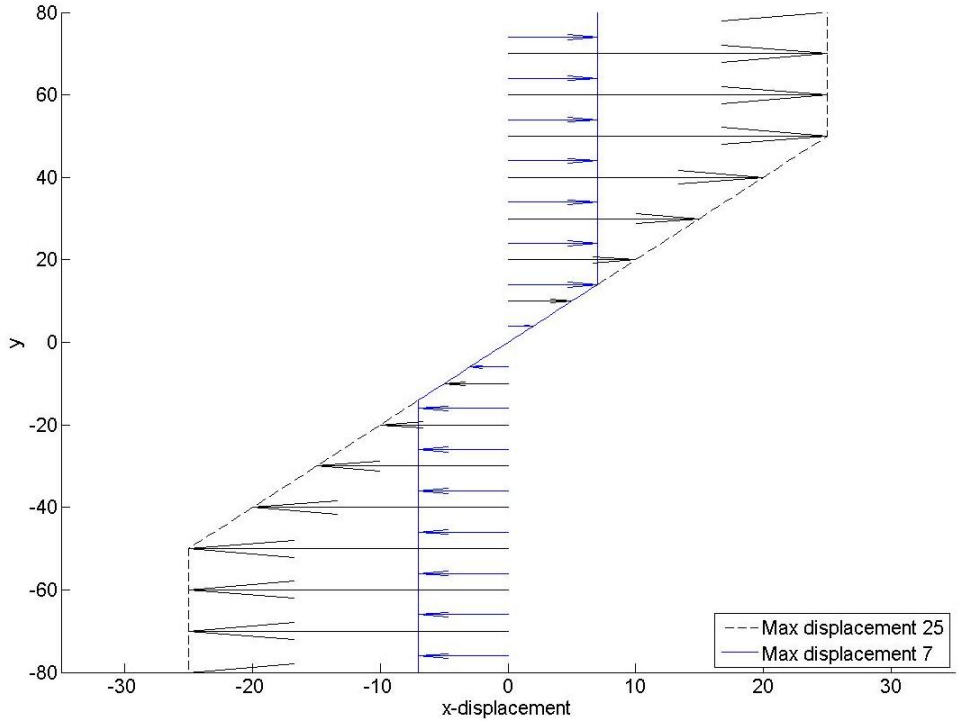

Figure 5. Uniform shearing velocity profile (gradient = 0.5) with velocities capped at ±25 px

A series of these images are processed using the current VB-PTV method with PIV guidance. In another test, the PIV guidance is also replaced with perfect guidance; that is, the characteristic distance σ is not interpolated from PIV results but is calculated based on a particle's location within the flow field. VB-PTV results from the region of shearing flow are shown in Figure 6. RMS error is shown in Figure 6a and the match yield and reliability at various gradient values are shown in 6b.

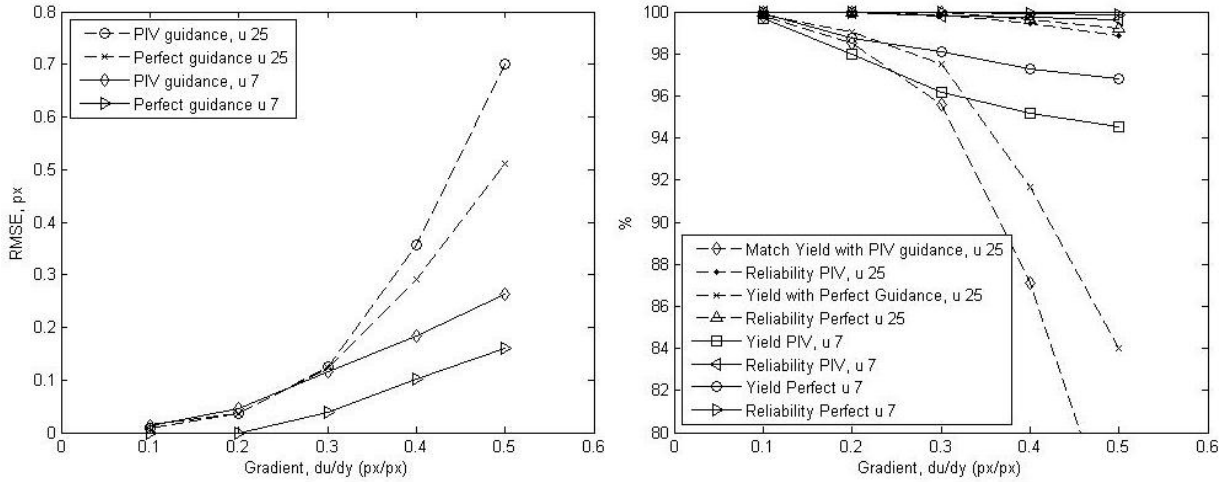

Figure 6 a) RMS Error, and b) percent yield and reliability vs. flow gradient for uniform shearing flow. Flows containing maximum displacements of ±7 and ±25 pixels are compared. The VB-PTV matching is guided both by PIV and exact analytic solutions.

We can draw some conclusions from the behavior shown in the figures above. Errors in the VB-PTV results grow with sharper gradients. At modest displacements, the errors rise in a roughly linear fashion but remain below 0.3 pixels. Larger displacements result in larger errors, especially as gradients increase above 0.3 px/px. In Figure 6b, we see similar behavior. The smaller displacements result in higher yield and reliability percentages in high gradient flow. Finally, it can be seen that PIV adds error to VB-PTV results and reduces the yield and reliability of matches. In these tests, the PIV method employed window-shifting and multipassing routines which resulted in 22 x 22 pixel interrogation windows. The error introduced by PIV guidance is

small at low gradients, and at the largest gradients adds between 0.1 and 0.2 pixels when compared with the VB-PTV results from using perfect guidance. In all cases, and even at high gradients, the reliability of matches remained at or above 99%. It should be possible to reduce the pixel-based gradients in experimental images by reducing the time step between exposures, though this can introduce other sources of error. In chapter 5, a simple modification to the current method is introduced as a means of reducing some of the errors seen in Figure 6.

4.4 Errors Due to Particle Image Density, Diameter, and Intensity

In addition to the images processed in section 4.2 which used a particle density of 0.03, moving wall images with densities of 0.01 and 0.06 were also generated and processed with VB-PTV. Match yield and reliability are higher at lower densities, and lower as the particle image density rises, as seen in Table 3. Errors are also lower at lower particle densities. Though the particle location identification algorithm is able to resolve particles which are overlapped by up to 50% of their diameters, at high particle image densities there are still many overlapped particles and clusters of overlapped particles which cannot be correctly identified and this contributes to the drop in performance. This is discussed further in section 4.7.

Table 3: Effect of particle image density on VB-PTV results

Particle Image Density	Particles Found	Matches Found	Match Yield with 1 Pixel Tolerance	Reliability with 1 Pixel Tolerance	Reliability with 0.5 Pixels Tolerance	RMSE (pixels)
0.06	4332	4021	91.3%	98.3%	93.4%	0.376
0.03	2865	2820	98.1%	99.6%	98.9%	0.111
0.01	1121	1116	99.5%	99.9%	99.8%	0.042

Plots of RMS error, match yield, and reliability versus particle image density are shown in Figure 7 below in order to describe optimum test conditions for the use of this VB-PTV algorithm. Lower particle densities produce less error and improve yield and reliability, while

higher particle image densities are desirable because they can provide information about the flow with higher spatial resolution. An optimum balance appears to be around an image density of 0.03 to 0.04. Even at high particle image densities, the reliability of matches remains above 98%.

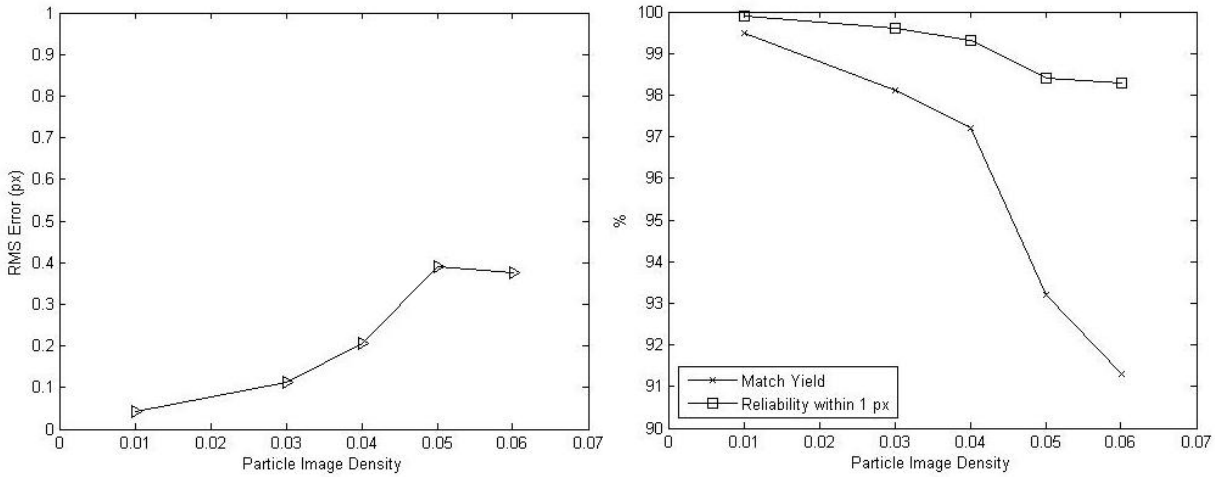

Figure 7. RMS error, match yield and reliability versus particle image density.

Particle images with large diameters provide more information to the particle identification algorithm which in theory allow for particle centers to be found more accurately. However, larger particle diameters also cause particle images to overlap more frequently and more severely than when small particle images exist. To find an optimum diameter, a moving wall image with 0.03 density is generated using various particle image diameters and processed with VB-PTV. Results are shown in Table 4. The reliability of matches when particle image diameters are 4 and 5 pixels remain above 95%, but drop to 86% when the diameter is increased to 7 pixels. The number of particles found by the algorithm also drops as particle images overlap more severely. Reliability of matches within 1 pixel remains above 96% for all cases.

Table 4: Effect of particle image diameter and intensity on VB-PTV results

Particle diameter	Particles Found	Matches Found	Match Yield with 1 pixel Tolerance	Reliability with 1 pixel Tolerance	Reliability with 0.5 pixels Tolerance	RMS (pixels
4	2865	2820	98.1%	99.6%	98.9%	0.112
5	2689	2585	94.9%	98.8%	95.4%	0.218
7	2412	2090	83.5%	96.3%	85.9%	0.391
Random Intensity and diameter	2792	2620	92.6%	98.7%	93.0%	0.259

As mentioned earlier, all previous synthetic particle images were generated using a constant peak intensity, I_o. In order to model experimental images, the peak intensity and particle image diameter are allowed to vary randomly. The peak intensities are distributed normally about a mean of 175 with a standard deviation of 25, and the diameters are distributed about a mean of 4 pixels with a standard deviation of 0.57 pixel. This randomization is performed separately for the first and second image in a sequence so that a particle may have a different appearance in frame 1 and 2. This randomization has a negative impact on the VB-PTV algorithm's performance and results in an error of 0.26 px, or twice the error for uniform particle images with 4 pixel diameters. It can be seen that even with the ability to resolve overlapped particle images, it is desirable to have smaller and sparser particle images for the greatest degree of accuracy.

4.5 Other Sources of Error

Two types of erroneous matches have been identified which affect particle tracking results. One is the result of pairs or triads of particle pairs which cross one another. The matching method is not always sensitive enough to choose the correct match out of two or more closely located particles, and the outlier detection routine may not identify these vectors as outliers since they don't vary greatly from nearby vectors. Such crossed vectors can account for as many as

half of all erroneous matches (those with an error greater than 1 pixel). An example is shown in Figure 8a. A simple routine is used to identify and either correct or remove crossed vectors. This process needs to be used selectively, as it assumes 2D incompressible flow, and may not be beneficial when correct matches should cross, as occurs, for example, in the VSJ images or in a highly rotating flow. When used on the moving wall or shearing flows it reduces RMS errors by 16%, on average.

A second major source of errors stems from the use of the particle location identification algorithm. In high density images, a cluster of overlapped particle images can easily contain four or more particles. Small changes in the relative location of these particles from one image to the next, due to flow gradients or a rotating flow, for example, can cause fairly different results in particle locations. It is even possible that overlapped particles which are identified as one particle in one frame can separate and be identified as two particles in another frame. An example of error due to particle location is shown in Figure 8b.

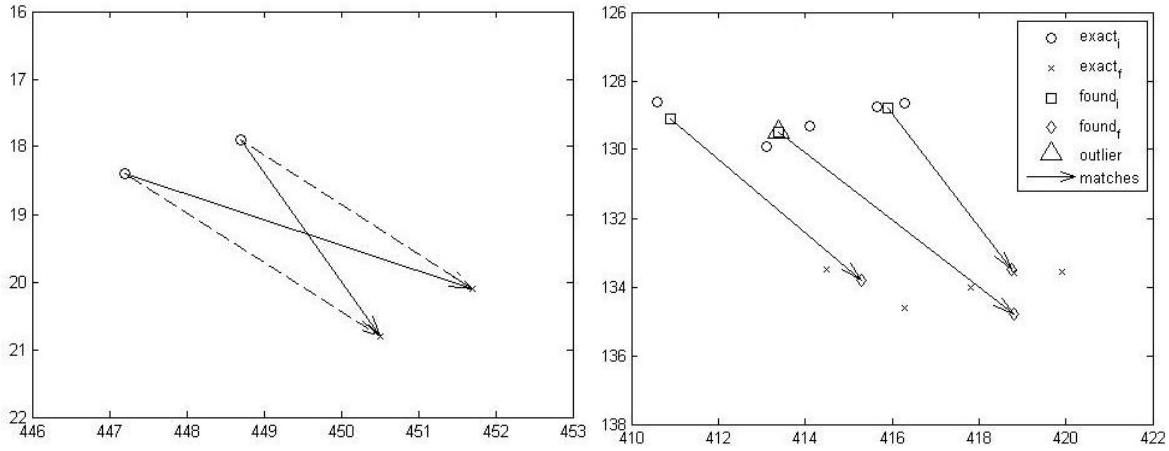

Figure 8 a) Example of error due to crossed matches. b) Errors caused by peak finding inaccuracy. Plot shows 5 pairs of exact particle peak locations and 3 pairs of peaks resolved by the particle location identification algorithm (particle diameter of 4 pixels).

4.6 VSJ STANDARD IMAGES

The current VB-PTV algorithm is used on standard images provided by the Visualization Society of Japan (VSJ). Results shown in this section are from series #301, which is generated using 3D large eddy simulation of a 2D planar jet impinging on a wall (Okamoto *et al*., 2000). The images are 256 x 256 pixels in size and contain around 4000 particle pairs with a maximum displacement of around 10 pixels. The particle images have a mean diameter of 5 pixels and a standard deviation of 1.4 pixels. The results are compared with the work of a number of other researchers who have used these images to test particle velocimetry methods (Ohmi and Li, 2000; Ruhnau *et al*., 2005; Mikheev and Zubtsov, 2008; Schindler *et al*., 2010; Brevis *et al*., 2011).

Table 5: VB-PTV results from VSJ 301 images and comparison to previous work

Algorithm	Particle Location	Matches Possible	Matches found	Matches Correct	Match Yield	Reliability
Present Work(Tracking only)	Known	4042	4039	3927	97.23%	97.15%
VAR(Ruhnau *et al*. 2005)	Known	4042	4039	3894	96.34%	96.41%
EPTV(Mikheev and Zubtsov, 2008)	Known	4042	3863	3823	94.58%	98.96%
ICCRM(Brevis *et al*.,2011)	Known	4042	NA	3980	98.46%	NA
Present Work (Particle ID+ Tracking)	Unknown	2095	1846	1761	84.06%	95.40%
EPTV(Mikheev and Zubtsov, 2008)	Unknown	2029	1759	1733	85.41%	98.52%
VAR(Ruhnau *et al*. 2005)	Unknown	NA	872	865	NA	99.20%
NRX(Ohmi and Li, 2000)	Unknown	NA	808	788	NA	97.52%
MF-EPS(Shindler *et al*., 2011)	Unknown	NA	1160	1146	NA	98.80%
2F-EPS(Shindler *et al*.,2011)	Unknown	NA	1123	1112	NA	99.00%

As with the moving wall tests, this experiment is divided into two parts. In the first, exact particle locations are known, and in the second, the particle location algorithm is used to identify particle centers before matching is performed. Images 0 and 1 from #301 series are used; they contain 4042 particle pairs which can be matched. When particle locations are known, 4039 matches are made, and 3927 of them are correct, which results in a match yield of 97.15% and a

reliability of 97.23%. When particle locations must be found, 2095 particles are identified, 1846 matches are made, and of these 1761 are correct within a 1 pixel tolerance, resulting in a match yield of 84.06% and a reliability of 95.40%.

These results are compared with the results available from other researchers in Table 5. When particle location are known, the current algorithm gives a match yield only slightly lower than the ICCRM algorithm used by Brevis *et al*. (2011), and the reliability of matches is comparable to other existing methods. The two methods with the best performance with known particle locations—the current algorithm and the ICCRM method—use cross-correlation results to improve tracking performance. In the ICCRM method, cross-correlation is used in a preprocessing stage of a relaxation method. For unknown particle locations, the current algorithm results in slightly lower yield and reliability percentages than other methods. It is, however, able to identify and match more particles than other listed methods, and a simple modification to the matching algorithm is introduced in chapter 5 which improves the yield and reliability values recorded in Table 5.

4.7 EXPERIMENTAL IMAGES

The current VB-PTV method is applied to experimental shear layer images. An interrogation area of 22 x 22 cm is tested, with flow velocities of 10.5 cm/s and 22.5 cm/s and a Reynolds number of 1.2×10^4. The images were 1000 x 1000 pixels. Details of the flow and experimental setup are described in Dabiri (2003). A vector field of the matching results is shown in Figure 9. The average of the free-stream velocities is subtracted from all vectors to better visualize the flow structures. 12429 and 12143 particles were identified in each frame of the image pair, and 7777 matches were found, resulting in a match yield of 64.0%. This is considerably lower than the yield when synthetic images with random intensities and diameters

were processed (see Table 4). The reason for this lies in the large number of rogue particles, or particles moving in and out of the laser sheet. To check the effect of rogue particles on the matching algorithm's performance, tests were performed on synthetic images with particles randomly removed and added. The degradation in match yield was approximately linearly related to the percentage of rogue particles. That is, when 10% of particles were randomized, the yield dropped by 10%; when 20% were randomized, the yield dropped by 20%, and so on. A visual inspection of the experimental shear images confirmed that around 20-25% of particle images in one frame could not be identified in the other frame, thus explaining the reduced match yield.

As an additional check on these results, experimental images were collected from uniform flow in a water tunnel with a velocity of 50 cm/s. The area imaged was 29 x 29 mm with a magnification of 0.22, and the flow was seeded with 44 micron particles. Few particles were observed moving in or out of the image plane. The match yield was 85.2% which is similar to the 92.6% yield found with synthetic images containing particle images with random diameter and intensity in Table 4. This suggests that the low yield in the shear layer images is due in large part to rogue particles, and that the current algorithm can be used to process experimental images.

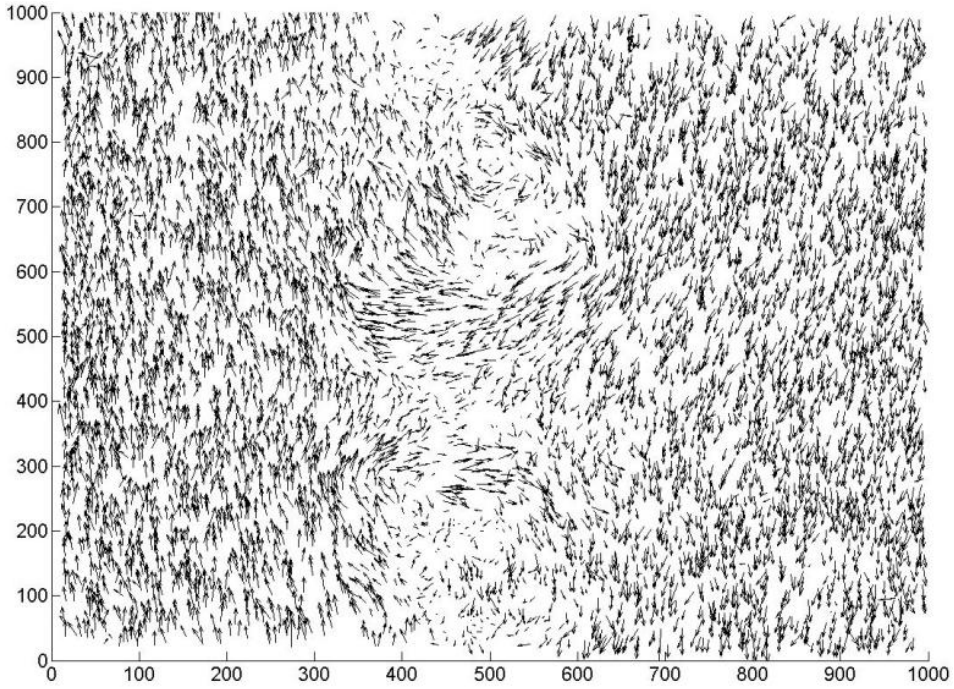

Figure 9. Results from experimental images of a shear layer flow with average of free stream velocity subtracted.

Chapter 5. AN IMPROVED METHOD OF PARTICLE TRACKING

It was seen that the current tracking algorithm's performance suffered in high displacement, high gradient flows. Figure 6 showed that errors increased and match yield and reliability percentages dropped as flow gradients increased, and this was more pronounced when the flow contained larger displacements. To address this we re-examine the Scott and Longuet-Higgins (1991) method of constructing a proximity matrix. The formula for the proximity matrix is repeated below,

(10)

This matching method was designed for computer vision, and might be used to match features in stereo images. Its strength lay in its simplicity and it required only one user defined parameter—the characteristic distance σ—which itself did not require a great deal of precision.

Results were roughly the same so long as σ was representative of the mean displacement between features, and preferably an overestimate of this value. In the particle tracking results recorded in chapter 4, σ was adapted to each target particle and was twice as large as the displacement found by interpolating the PIV velocity field to a particle's location. Scott and Longuet-Higgins explain that features in successive images will often be related by a transformation which is affine or nearly so. They show that when one set of points is mapped into another by a translation, expansion, or shear deformation the 1:1 mapping minimizes the sum of squares of distances between the sets of points. And they argue that by choosing a sufficiently large σ, their matching method possesses this same property and thus is useful in discerning the transformations which exist between feature sets in many and varied image pairs. Here, a more graphical view is taken of the matching method, specifically in the construction of the proximity matrix. The process of performing singular value decomposition on this matrix to obtain a pairing matrix P_{ij} is unchanged and so the properties which allow for robust matching remain.

5.1 Modifying the Proximity Matrix

We can imagine a Gaussian "proximity surface" existing around each target particle j in frame 1, described by Equation 10, which can be thought of as a measure of a location's proximity to a target particle. This surface reaches a maximum of 1 when r_{ij} is zero and decreases monotonically as r_{ij} increases. A candidate particle i in frame two will have some value depending on where it falls on this proximity surface. This value is element G_{ij}. Since the proximity matrix is constructed using the principle of proximity, candidate particles which are closer to the target particle (that is, they experienced a smaller displacement) will have a larger element G_{ij}. If the proximity matrix were used on its own for matching, it would result in a

nearest-neighbor method. The parameter σ controls the rate of decay or the breadth of this Gaussian surface. When σ is small, the surface is small and only candidate particles near the target particle will have large values in the pairing matrix *G*. When σ is large, the Gaussian surface widens and more candidate particles will have large values. The process of creating a pairing matrix *P* permits the correct match to be found in most cases because matching is performed on all particles at once and coherent particle motion is preferred to simply matching a target particle with its nearest candidate. However, reliance on this formulation of the proximity matrix introduces unnecessary noise to the matching process, especially for large displacements. Two simple modifications are introduced here.

Because an estimate of local displacements can be obtained from PIV, a more selective proximity matrix can be constructed. In the first modification, the proximity matrix is created using the following formula:

(11)

Instead of favoring candidate particles with zero displacement ($r_{ij} = 0$), the new proximity surface reaches a maximum when r_{ij} is equal to the displacement predicted by PIV, r_{PIV}, resulting in a ring-like proximity surface. In the second modification, the expected direction as well as displacement is taken into account.

(12)

The terms dy_{ij} and dx_{ij} are the pixel distances between target particle *i* and candidate particle *j*. The dy_{PIV} and dx_{PIV} terms represent the *x* and *y* displacements of target particle *i* predicted by PIV. The second exponential term in Equation 12 provides a comparison between a candidate particle's angle in relation to the target particle and the PIV prediction of the angle of a

target particle's displacement. The resulting proximity surface contains a peak in each quadrant, one of which will lie at the expected location of the target particle's matching candidate particle. The scaling parameters ε and φ are discussed later. For comparison, proximity surfaces for a predicted displacement of 7 pixels at 45 degrees are shown in Figures 10, 11, and 12.

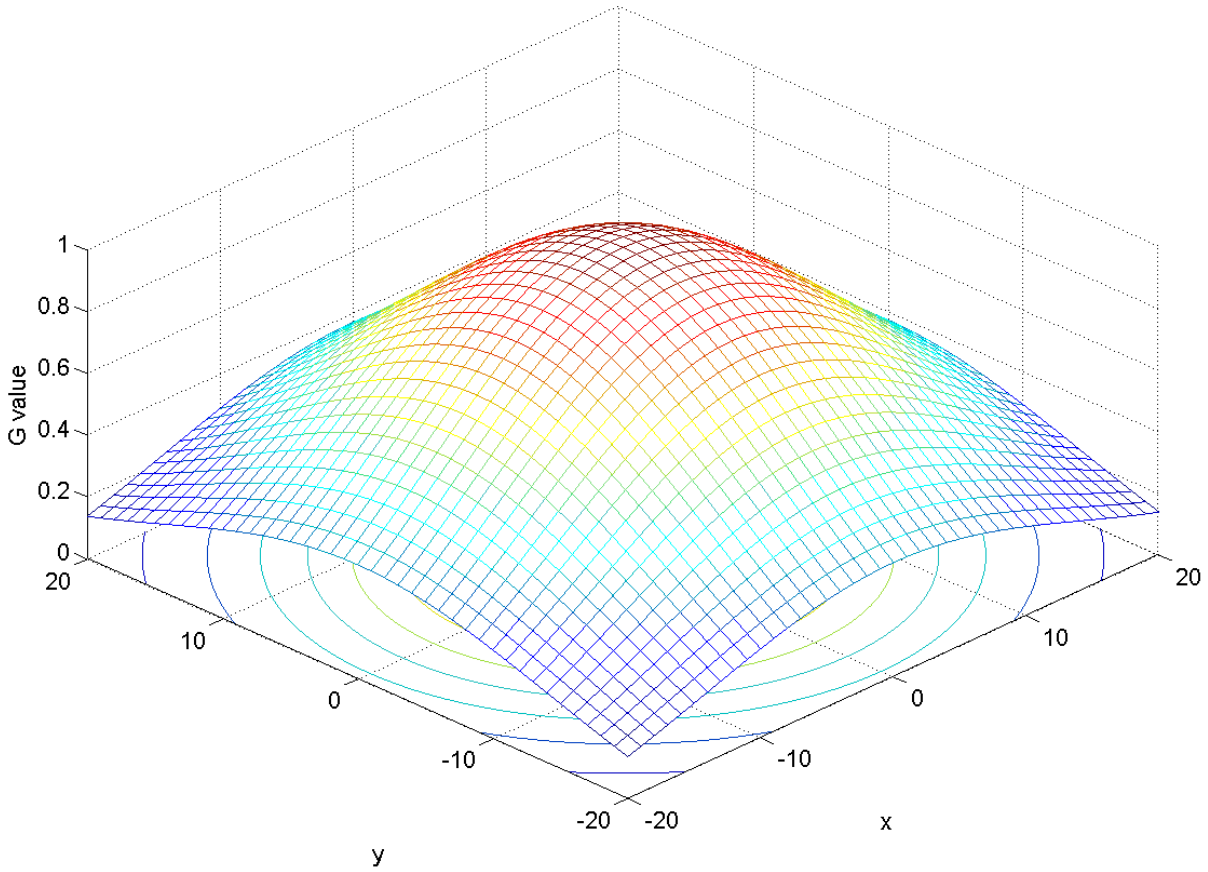

Figure 10. Proximity surface about a target particle using Scott's and Longuet-Higgins' proximity matrix

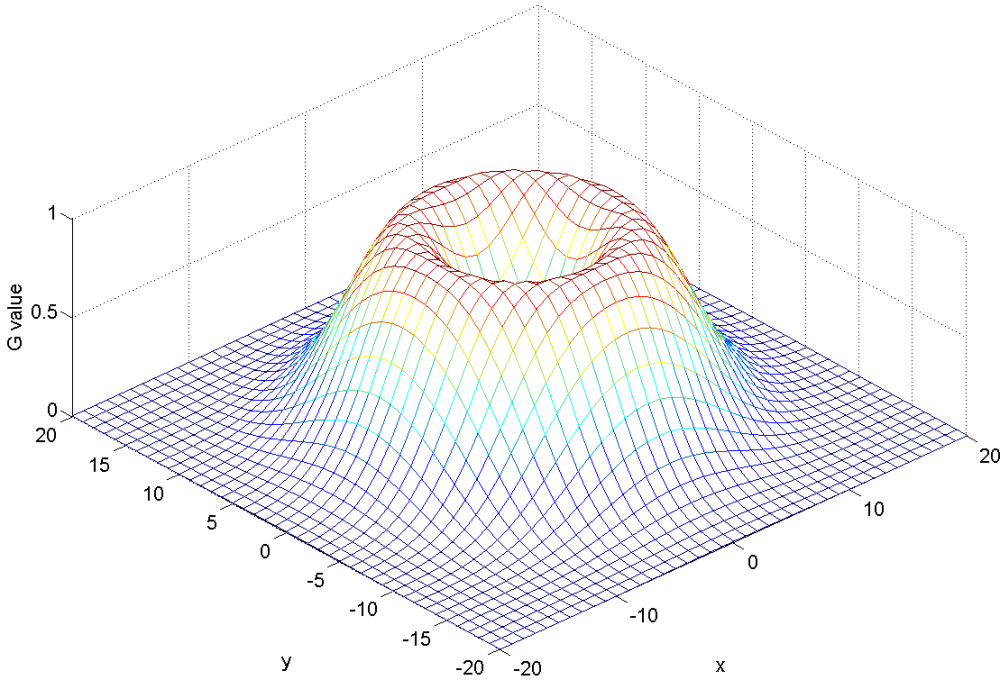

Figure 11. Proximity surface about a target particle using first modified proximity matrix formula

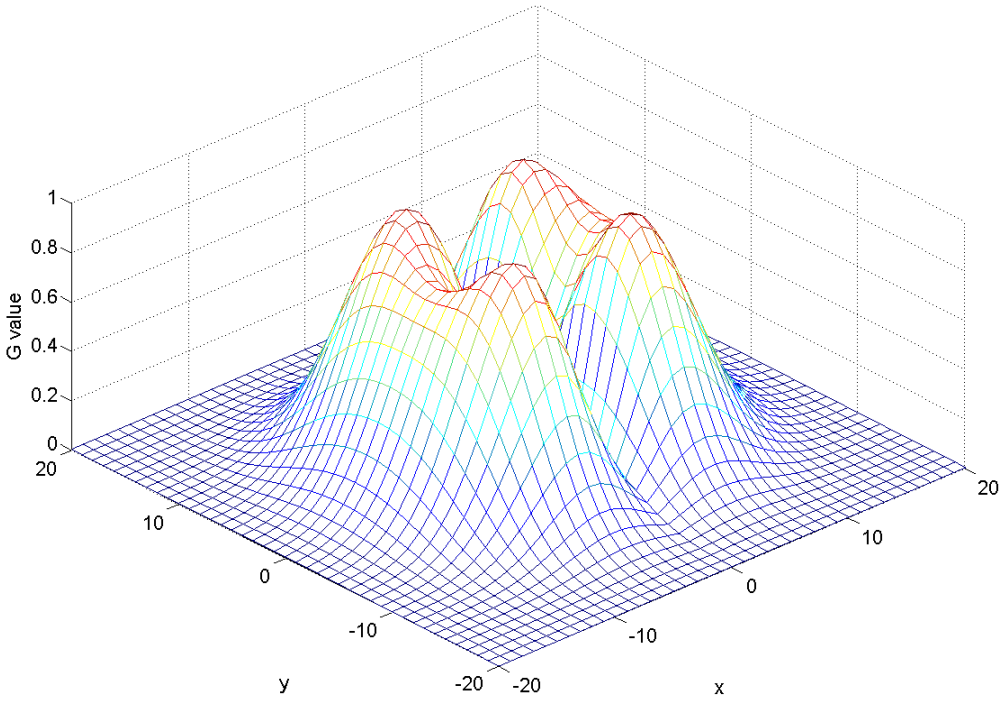

Figure 12. Proximity surface about a target particle using second modified proximity matrix formula

It can be seen in the above figures that the original method can result in many candidate particles with large values in the proximity matrix, while the modified methods are far more selective and will result in a more sparse proximity matrix. This becomes more pronounced with larger predicted displacements. Like the original Gaussian proximity surface, the ring-like shape of the first modified method and the peaks in the second method can be widened or narrowed by controlling the parameters ε and φ. In the following results, ε was set to a value of 4. This was found to be the optimum value for high gradient flows, and performed acceptably well in all other experimental flows. If this parameter were made to be adaptive, it might be related to the estimated error between the predicted displacement, as found by PIV, and the true particle displacement. The parameter φ is set to unity. The proximity matrix created using Equation 11 or 12 is still combined with the correlation term described in section 2.1 before it undergoes singular value decomposition.

5.2 Results of Modified VB-PTV Performed on Synthetic Images

These modified methods were used to process many of the same synthetic flows described in section 2.1. Since the modifications were initially produced in response to errors seen in VB-PTV performed on high gradient flow, results from uniform shearing flows (see section 4.3) are presented first.

It can be seen in Figure 13 that the modifications have little effect on RMS error, match yield, and reliability when compared with the original matching algorithm's performance on shearing flow with displacements of up to ±7 pixels. The improvement can be seen by looking at Figure 14, which shows results from shearing flows with displacements of up to ±25 pixels. Both modifications lower error and increase yield and reliability, and the second modification provides the most improvement. As much as 60% of the RMS error is removed by the second

modification, and match yield is increased by up to 24%. Even the already high reliability percentage is improved with these modifications, only dropping to 99.6% at a gradient of 0.5 px/px.

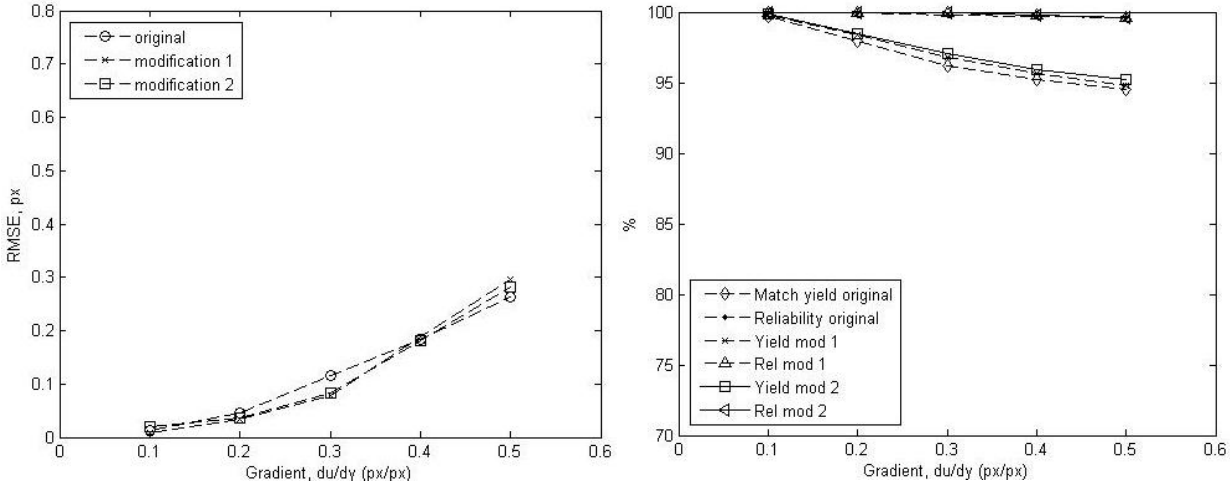

Figure 13 a) RMS error versus flow gradient with a maximum velocity of 7 pixels, and b) match yield and reliability percentages versus gradient

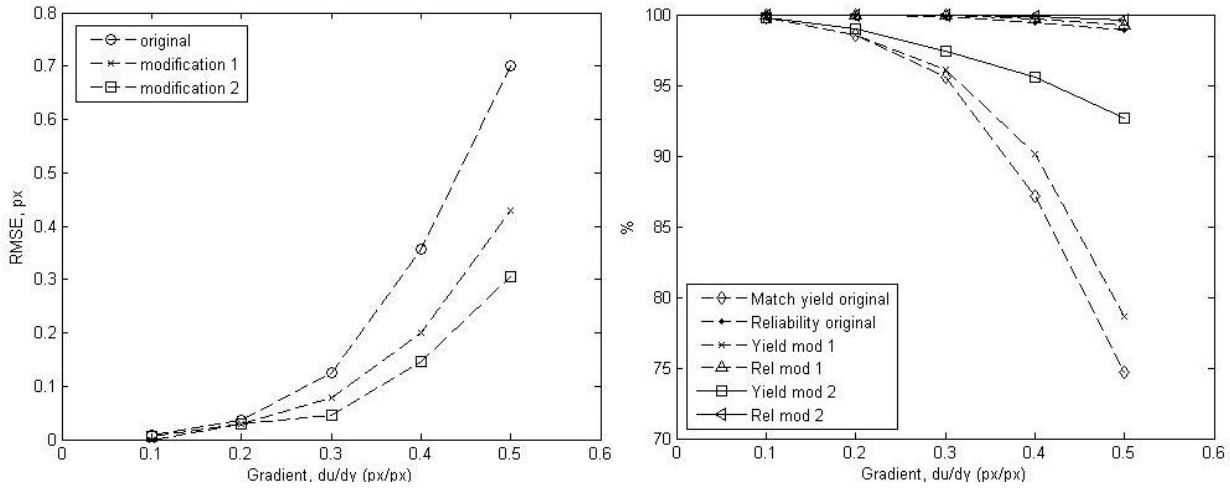

Figure 14 a) RMS error versus flow gradient with a maximum velocity of 25 pixels, and b) match yield and reliability percentages versus gradient

The second modification results in an error curve which resembles that for the small displacement shearing flow, with a maximum RMS error of roughly 0.3 px at high gradient values. It would therefore seem that this modification reduces the error induced by large

displacements in shearing flow, though not the errors induced by high gradients themselves. Given these improvements, the modified matching methods are applied to other synthetic images with more complicated flows.

Synthetic images generated using a moving wall flow, or Stokes' first problem, are processed using the modified matching. The flow parameters are U=10, μ = 5, and t=75 and flow displacements vary from 0 to 10 with maximum gradients of 0.5 px/px. Both the shearing flow and moving wall flow are unidirectional, so to fully utilize the directional guidance provided by the second modification, synthetic images are generated using a 2-dimensional Oseen vortex flow described as

$$\frac{\qquad}{\qquad} \qquad (13)$$

Radial velocity is zero, Γ=5000π, γ=5000, r is the radial distance from the center of the vortex. Displacements in this flow vary from 0 to 22 pixels and gradients $\left(\partial u_\theta / \partial r\right)$ vary from near zero to 0.5 px/px. The RMS errors from these flows are presented as bar charts in Figures 15 and 16. Three cases are shown for each matching method: one with the typical PIV guidance and known particle locations; one with known locations and perfect guidance using the analytic flow solutions to guide matching; and one with unknown particle locations and regular PIV guidance. The match yield and reliability percentages were generally unchanged or slightly improved by the modifications.

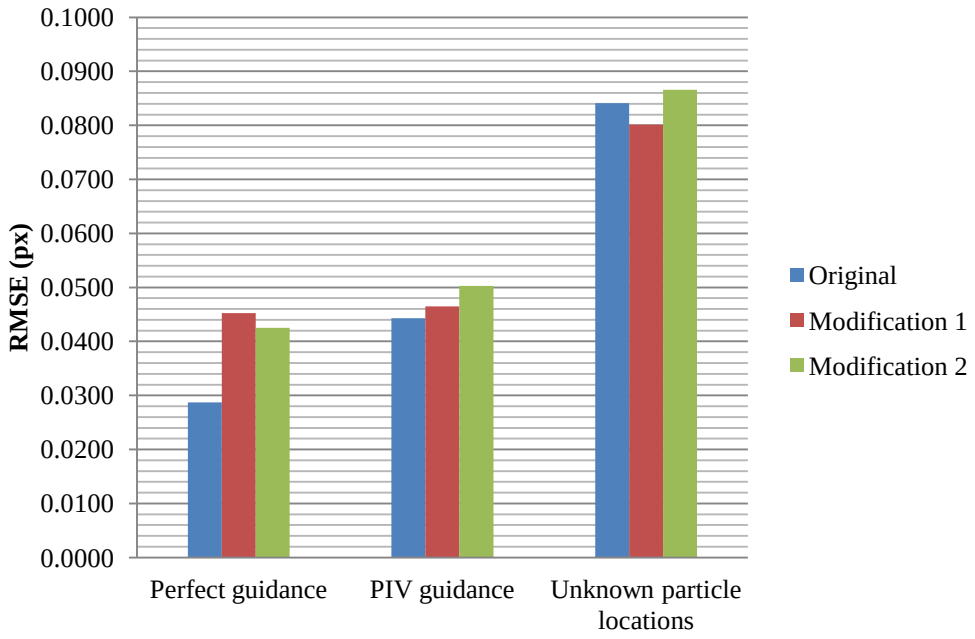

Figure 15. RMS error of VB-PTV results for a moving wall flow

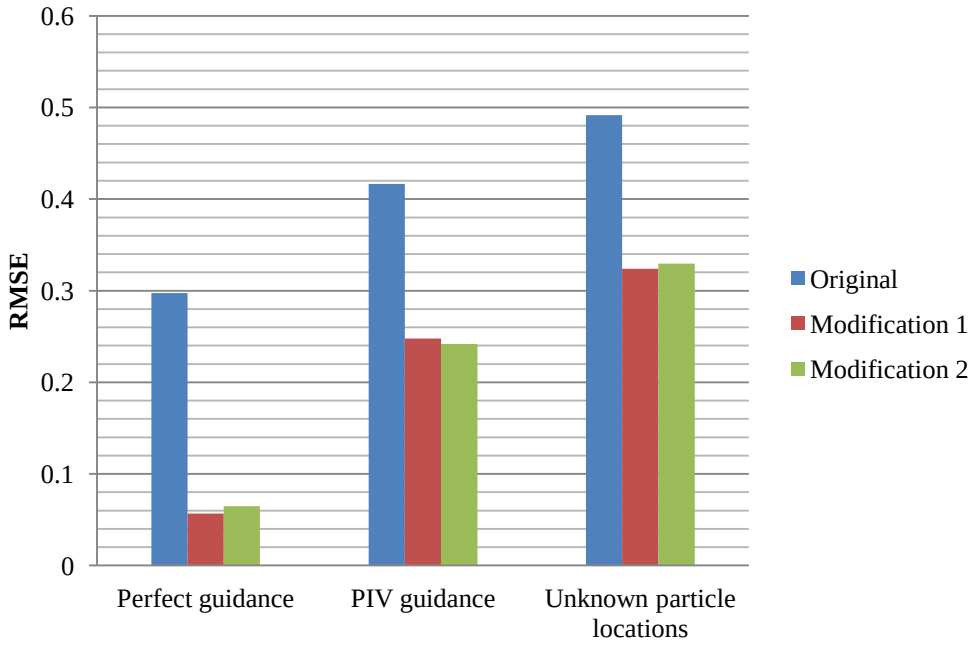

Figure 16. RMS error of VB-PTV results for a 2D Oseen vortex flow

The results show that for the moving wall flow, the modifications to matching provide no benefit or may result in slightly higher errors. The moving wall flow contains only a small region with high gradients, and nowhere do displacements above 10 pixels exist. The majority of

particle locations change either not at all or negligibly between the image frames. For this reason it would not be expected that there would be a large difference between the original method, which favors a close match to a distant one, and the modified methods. However, a dramatic improvement in performance can be seen in the Oseen vortex flow, which contains widely varying gradients, displacements, and flow directions and so could be considered a better gauge of VB-PTV performance on a generic flow.

The modifications to the matching algorithm reduce the RMS error for an Oseen vortex flow by as much as 80% in the idealized case of perfect guidance, by more than 40% when PIV guidance is used with known particle locations, and by roughly 33% when particle locations must be identified. The final test is performed on the synthetic images provided by the Visualization Society of Japan (see section 4.5). The images are reprocessed and the results of tests with known particle locations are shown in Table 6 compared with the original method and the results from other existing PTV methods. Both modifications result in improved performance when compared with the original matching method. Both the match yield and reliability are improved to over 99%.

Table 6.Comparison of modified matching with original results and existing PTV techniques

Algorithm	Matches Possible	Matches found	Matches Correct	Match Yield	Reliability
Present Work (Original method)	4042	4039	3927	97.23%	97.15%
(First modification)	4042	4038	4002	99.01%	99.12%
(Second modification)	4042	4032	4014	99.31%	99.55%
VAR(Ruhnau *et al*. 2005)	4042	4039	3894	96.34%	96.41%
EPTV(Mikheev and Zubtsov, 2008)	4042	3863	3823	94.58%	98.96%
ICCRM(Brevis *et al*.,2011)	4042	NA	3980	98.46%	NA

B

ased on the results seen above these modifications are able to improve the matching results, in terms of reduced error and improved match yield and reliability, for various flow types. It is not

clear from these results whether the first or second modification will result in greater improvement in a general application, but it is recommended that one of these modifications to the matching algorithm be used in any future applications of VB-PTV.

Chapter 6. POST-PROCESSING OF PTV DATA

In this chapter, a natural neighbor-based interpolation method is described and tested using synthetic PTV data. These results are compared with three other commonly used techniques for interpolation: one grid-based and two non-grid-based methods. Additionally, a smoothing technique from computer graphics is applied to artificial noisy PTV data. Errors in strain rate estimates from all interpolation methods are compared when using perfect data, unsmoothed noisy data, and smoothed data. The interpolation methods are also compared using their computation time and ability to resolve small flow features. Recommendations on the suitability of this natural neighbor-based interpolation are made.

6.1 METHODS OF INTERPOLATION

As discussed in section 1.2, velocity measurements are typically a means of obtaining other flow information, like vorticity or strain rates. A key advantage of PIV over PTV in this respect is that velocity data is found on a regularly spaced grid and so calculating derivatives can be as simple as implementing a central differencing scheme. Since PTV tracks individual, randomly located particles, the velocity field is scattered and finding its derivatives becomes more difficult. Here, four methods of interpolation are briefly described: a new natural neighbor-based method of interpolation, and the established methods of adaptive Gaussian windows, radial basis functions (RBF) and kriging interpolation.

6.1.1 *Natural Neighbor-Based Interpolation*

Scattered data interpolation is hardly exclusive to particle velocimetry applications, and methods have been developed for use in fields as varied as cosmology (Bernardeau and van de Weygaert, 1996; Schaap and van de Weygaert, 2000) and geostatistics (Journel and Huijbregts, 1978). We can arrive at another method for deriving differential quantities by manipulating a technique developed in the field of structural modeling. The Natural Neighbor Galerkin method, or natural element method (NEM), is a technique which can be used to solve elliptic PDEs (Sukumar, 2001), and has the desirable trait of being meshless, meaning that it does not require a structured grid of interpolation nodes (Gonzalez *et al*., 2004). Natural neighbor interpolation is based on a Voronoi tessellation of scattered nodes. A field of randomly located nodes can be divided into Voronoi cells, which are related to Delaunay triangles, but can simply be thought of as the region around a node point which is closer to that point than to any other node. Figure 17 displays a Voronoi cell about a point *p*. Natural neighbors can be defined as the set of nodes around a point of interest whose Voronoi cells share a face with that point of interest's cell. In Figure 17, solid lines connect point *p* with its natural neighbors.

Two natural neighbor-based interpolation methods were developed: Sibsonian (Sibson 1980) and non-Sibsonian, or Laplacian, interpolation (Christ *et al*., 1982; Belikov *et al*., 1997; Hiyoshi and Sugihara, 1999). At the heart of the interpolation method is a weighting term which determines the amount of influence a neighbor has on the interpolated value at the point of interest. The Sibson interpolant uses an area-based weighting term whose details can be found in the original paper or Sukumar (2001). The non-Sibsonian or Laplace interpolant (hereafter referred to as the Laplace interpolant, per Hiyoshi and Sugihara, 1999) is weighted with distances between neighbors and Voronoi cell edges, rather than areas, and so has the advantage

of being more computationally efficient (Sukumar 2001). A brief outline of this method of interpolation in 2 dimensions is provided below.

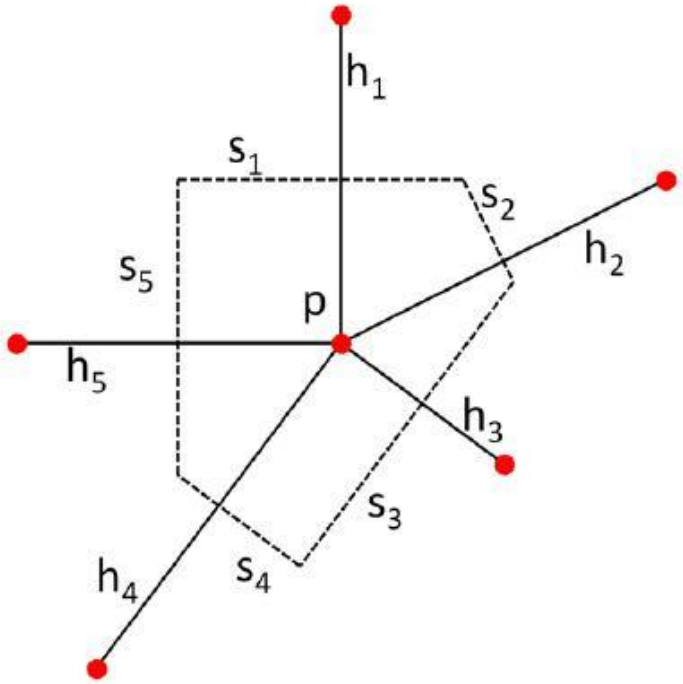

Figure 17. Voronoi cell and natural neighbors of point p (Duncan 2009).

First, a naming convention is established using the diagram in Figure 17. The distance between point p and a natural neighbor is h_i, for $i=1,2...n$ where n is the number of natural neighbors of point p. The Voronoi cell edge which is shared by point p and natural neighbor i is named s_i. The ratio of these distances will form a weighting term used in the interpolation. If we wish to interpolate a function ψ to point p, we use the following formula:

(14)

Here φ is the weighting of each natural neighbor based on the ratio of s and h:

(15)

(16)

Combining these formulae gives us the interpolation of a function ψ in terms of *s* and *h*. It can be seen that this interpolation method requires only simple algebraic calculations:

(17)

This method of interpolation was adapted for use in estimating differential terms within a PTV velocity field by Duncan (2009). The desired gradient function ψ is unknown at each velocity data point, but it can be estimated by also approximating the gradient (in this example, $\partial u / \partial y$) at each natural neighbor using the following formula:

(18)

In-house work at the University of Washington (Duncan 2009) suggested that this method of interpolation could give very accurate results when used to calculate strain rate in various synthetic flows, outperforming interpolation performed using a radial-basis function (RBF). The RBF method was used to interpolate scattered velocity data to nodes spaced at twice the mean particle spacing of the velocity datasets and gradients were calculated using a central differencing scheme. (This is a less than ideal method of using RBF-based interpolation, as will be seen later). However when just 1% noise was added to the velocity data, the RMS error of strain rate estimates increased by a factor of 22 for the natural neighbor-based interpolation. If

the natural neighbor method was a viable option for interpolation of scattered data, this sensitivity to noise would need to be addressed. Though "many smoothing operations" (Duncan 2009) were used to remove noise from the velocity field, errors in strain rate estimates remained high. It was not clear what smoothing routines were used to clean the velocity fields.

6.1.2 *Adaptive Gaussian Window*

The three methods of interpolation selected for comparison to the natural neighbor-based (NN) method are the common adaptive Gaussian window method, RBF interpolation, and kriging interpolation. Each is described briefly here.

The adaptive Gaussian window (AGW) technique uses a Gaussian weighting function to interpolate velocity data to a series of regularly spaced nodes. Interpolating to determine a function $F(x,y)$ using some number k scattered data points, f_k, is accomplished using the following formulae (Spedding and Rignot, 1993):

$$\frac{\sum_{k=1}^{n} w_k f_k}{\sum_{k=1}^{n} w_k} \qquad (19)$$

$$w_k = e^{-\left(\frac{r_k}{\sigma}\right)^2} \qquad (20)$$

The Gaussian weighting function w_k is a function of the distance, r_k, between the interpolation node and velocity data, and the scaling factor σ is set to $1.24 \cdot mps$, where mps is mean particle spacing, $1/\sqrt{\pi p}$, and where p is the particle image density, or number of particle images per square pixel. The interpolation node spacing was set to be $1.5 \cdot mps$. While this method can be used globally, the inverse distance weighting means that velocity data points far from an interpolation node have virtually no influence on the interpolated value. For this reason

and to improve computational efficiency, the data field is broken into interpolation windows around each interpolation node and so practically acts as a local interpolation.

6.1.3 *Radial Basis Function (RBF)*

The radial basis function method of interpolation examines the global data field and finds a function which matches known data points. The function is assumed to have the form (Chirokov, 2006)

(21)

Here n is the number of data points, which are located at x_i. The radial based function, φ, can take various forms. Once the coefficients have been estimated, the function can be sampled at any point. In the current work a cubic basis function is used which has the form

(22)

The estimated function is sampled at every pixel location and the gradients calculated using these interpolated velocity values. Many more details on RBFs can be found in, for example, Buhmann (2003), and details on the specific RBF Matlab® toolbox used in this work can be found in Chirokov (2006).

6.1.4 *Kriging Interpolation*

The final method of interpolation used for comparison with the NN technique is kriging interpolation. The focus of this chapter is the NN interpolation method's performance, and since even a brief treatment of kriging interpolation quickly becomes a long list of equations and variable definitions, interested readers are referred to Gunes *et al*., (2006); Sacks *et al*., (1989); and Lophaven *et al*., (2002), which describe the kriging method used in this section. Like the RBF

method, a model function is created which estimates the data surface and can be sampled at any location within the data field and gradients calculated from these sampled points. The DACE toolbox for Matlab® is employed, using a quadratic regression model based on second order polynomials, and a spherical correlation model. Kriging has been shown to be a useful method of interpolation, though it does require manipulation of several potentially large matrices and so can become computationally expensive.

6.2 SYNTHETIC FLOWS USED FOR TESTING

In Duncan (2009), six synthetic flows were generated and tested using NN and RBF interpolation: Uniform flow, shearing flow, solid body rotation, 1-D Oseen flow, flow with a non-smooth strain rate profile, and a series of 2-D vortices. Here the uniform flow is dropped as it revealed little about the relative strengths of the interpolation methods. The flow profiles are briefly described below.

The uniform strain rate flow has a gradient of $\partial u / \partial y = 0.05$. The solid body rotation (SBR) flow has no strain rate but a uniform velocity of 10° in the angular direction. The 1D Oseen vortex is described by

(23)

where $B=2$ and $\Gamma=100$. The non-smooth strain rate velocity field is designed with the intention of demonstrating the ability of the NN method to recreate even sharp changes in gradients because it only examines local velocity information. It is described as

(24)

Here Δ is the step width, or can be thought of as the width of the region within which the velocity increases from 0 to 2. In most examples Δ is 18 pixels, unless otherwise stated. The 2D vortices are described with the following:

(25)

(26)

The strain rate profiles of these last three flows are shown below in Figure 18.

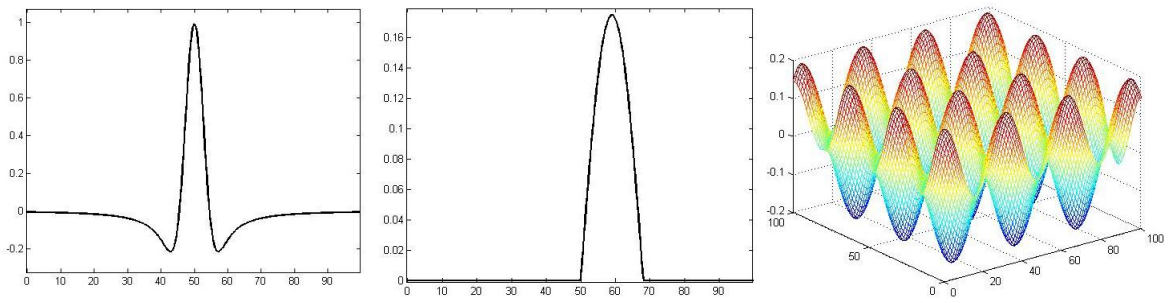

Figure 18 Strain rate profiles for a) 1D Oseen flow, b) non-smooth strain rate flow and c) 2D vortices

6.3 Interpolation Results

Each of the four interpolation methods is used to estimate the strain rate, given as

(27)

All synthetic data sets consisted of 300 data points within a 100 x 100 pixel region. The RMS error between the found and exact strain rate values are averaged over 100 data sets. Results from velocity fields with no noise are recorded in Table 7. Cells shaded gray indicate the interpolation scheme which provided the best performance for a given synthetic flow. In addition

to the averaged RMS errors recorded in Table 7, the standard deviation of the RMS errors was calculated for each test, and this value was considered the uncertainty interval of that dataset. When the uncertainty interval of the method with the lowest RMS error overlapped with that of another interpolation scheme for a given flow type, that interpolation scheme was considered to have given an estimate of the strain rate in a flow whose error was statistically similar to the method with the best performance. The values in these cells are in a bold underlined font.

Table 7.RMS error of strain rate from various interpolation schemes on synthetic velocity fields without noise

Flow Description	Natural Neighbors	RBF	Kriging	AGW
Uniform strain	2.09E-03	**1.42E-15**	**4.32E-16**	1.38E-02
Solid body rotation	**2.51E-13**	**4.79E-15**	**1.40E-15**	5.58E-02
1D Oseen	4.96E-02	8.11E-02	**1.96E-02**	1.14E-01
Non-smooth strain	7.37E-03	**5.26E-03**	**3.86E-03**	2.30E-02
2D vortices	3.41E-02	**1.32E-02**	**1.78E-02**	4.69E-02

It can be seen from these results that RBF and kriging interpolations result in the lowest errors in strain rate estimates. The AGW method is never as accurate as the non-grid-based interpolations. The NN interpolation only performed as well as RBF and kriging in solid body rotation flow. This would seem to contradict the findings in Duncan (2009) in which NN interpolation outperformed RBF interpolation with noiseless data. However, as was suggested earlier, the RBF interpolation was used like a grid-based interpolation method, with a widely spaced grid of sampling nodes. Since RBF (and kriging) result in a continuous function throughout the data field, it is possible to sample them at every location, greatly reducing the

gradient estimate errors. To help visualize the comparative performance of these interpolations, example results from the non-smooth strain rate flow are shown in Figure 19. Since the flow is unchanging in the x-direction, the strain rate estimates are collapsed onto one plane for ease of viewing.

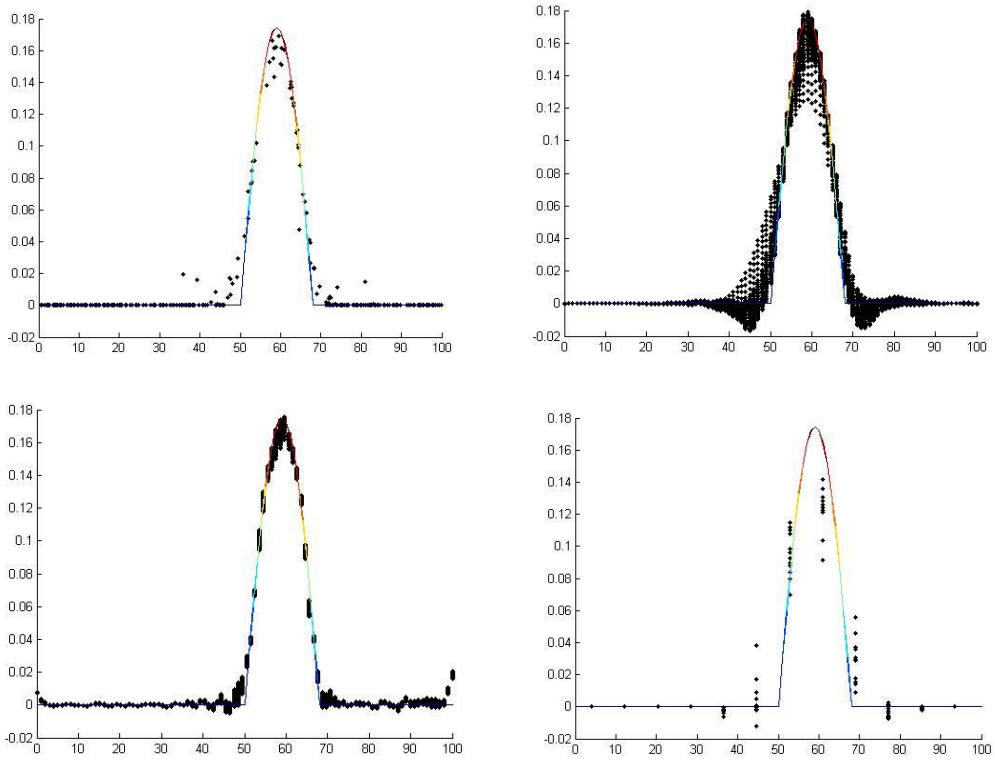

Figure 19. Strain rate estimates of the non-smooth strain rate flow, clockwise from top left a) natural neighbor interpolation, b) RBF, c) adaptive Gaussian window, d) kriging

It can be seen in these examples that, while NN interpolation is able to follow the sharp jump in strain rate without too much smoothing, it does suffer from outliers, even with perfect data. The RBF interpolation experiences some overshoot at the jump in strain rate, while kriging follows the strain rate behavior quite well. The AGW method smooths the peak of the strain rate feature due to its grid-based interpolation scheme which averages a group of velocity data points.

In spite of this performance, a better understanding of NN interpolation's performance was desired and so these interpolations were repeated on noisy velocity data sets. PTV error was simulated by recreating particle image location errors. A real PTV image could also include spurious vectors which were not detected by outlier detection schemes, but this source of error was not modeled. Data sets were simulated as a series of matched particle coordinates from which displacements could be calculated. Random position error was added to these particle locations by simply shifting the coordinates some distance in a random direction. The mean magnitude of this shift was set to 0.2 pixels with a standard deviation of 0.067 pixels (see Figure 7.13 in Adrian and Westerweel, 2011). The results from interpolating these noisy data fields are shown in Table 8.

Table 8. RMS error of strain rate from various interpolation schemes on synthetic velocity fields with noise added

Flow Description	NN	RBF	Kriging	AGW
Uniform strain	4.21E-01	1.14E-01	**3.19E-02**	**2.82E-02**
Solid body rotation	4.01E-01	1.26E-01	**3.07E-02**	5.87E-02
1D Oseen	4.57E-01	**1.48E-01**	2.19E-01	**1.17E-01**
Non-smooth strain	4.44E-01	**1.26E-01**	7.58E-02	**3.46E-02**
2D vortices	3.95E-01	**1.31E-01**	**7.38E-02**	**5.54E-02**

The best performance now comes from the AGW interpolation, where its inherent smoothing tendencies allow it to be less affected by the random noise. Data sets with mean noise magnitudes of 0.1 and 0.3 pixels were also processed. More noise resulted in overall higher errors, and less noise in less error, but the relative performance of the interpolation methods

remained the same. Plots of the results from a non-smooth strain rate profile are once again informative and shown in Figure 20.

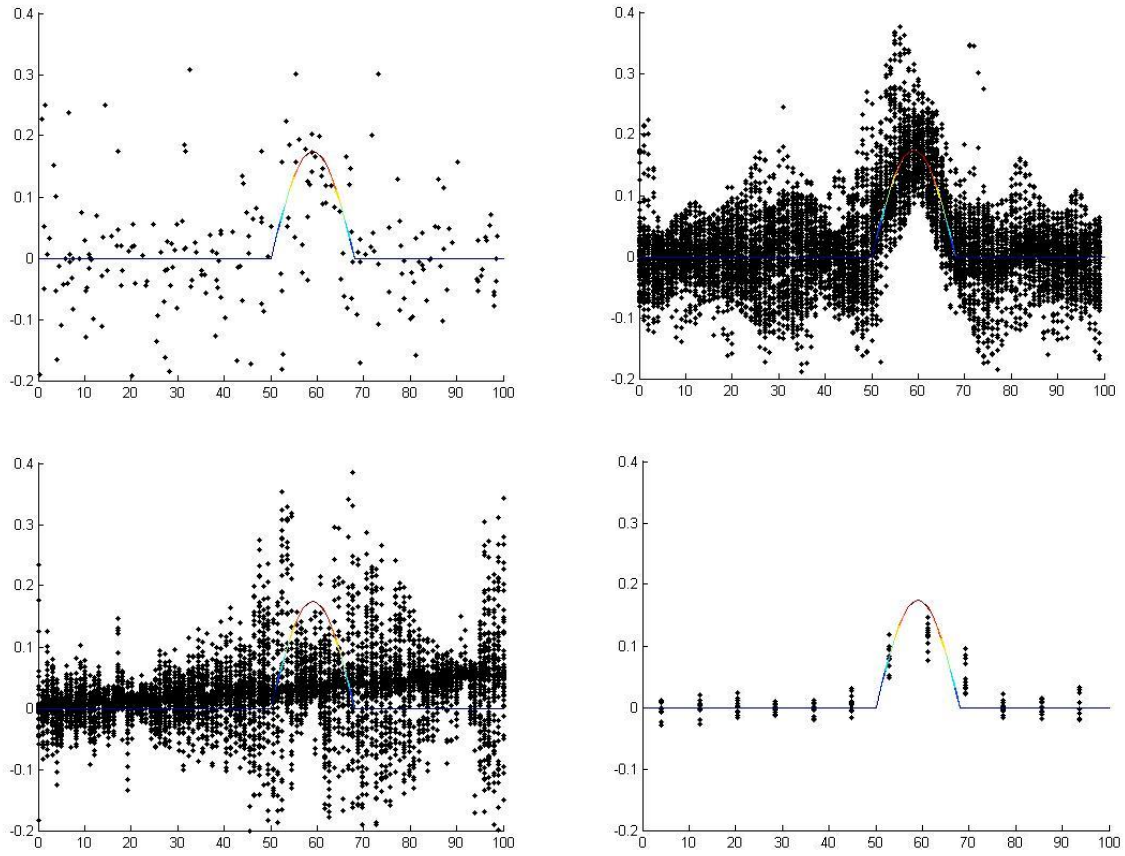

Figure 20. Strain rate estimates of the non-smooth strain rate flow with noise, clockwise from top left a) natural neighbor interpolation, b) RBF, c) adaptive Gaussian window, d) kriging

This random noise causes the strain feature to be essentially lost in the NN and kriging methods. RBF somewhat follows the strain profile, but it also displays a great sensitivity to noise. The AGW method appears to follow the strain profile approximately as well as it did with perfect data. Some way of addressing the severe drop in performance must be devised in order to use any of the non-grid based methods with experimental data.

Towards that end, a technique of smoothing randomly located data is adopted from computer graphics which achieves local surface area minimization. It has the desirable traits of

1) being a local smoothing which preserves actual feature edges, if used properly, and 2) of being simple to implement with multivariate data, such as two velocity components existing on a 2-manifold. Details of the method can be found in Desbrun *et al*., (1999 and 2000). After random noise was added to the synthetic data sets, it was removed using this feature-preserving denoising technique, and the datasets reprocessed. Denoising was not used in conjunction with the AGW method, since that grid-based technique already results in smoothing of noise, and it was found that additional denoising only increased RMS error. The results are shown in Table 9.

Table 9. RMS error of strain rate from various interpolation schemes on synthetic velocity fields with a denoising technique applied

Flow Description	NN	RBF	Kriging	AGW
Uniform strain	8.16E-02	**3.10E-02**	**2.05E-02**	2.85E-02
Solid body rotation	7.88E-02	3.09E-02	**2.13E-02**	5.89E-02
1D Oseen	**1.69E-01**	**1.28E-01**	**1.29E-01**	**1.18E-01**
Non-smooth strain	8.58E-02	**3.18E-02**	**3.60E-02**	**3.47E-02**
2D vortices	9.83E-02	**5.05E-02**	5.78E-02	**5.54E-02**

Application of this denoising scheme once again results in the best performance coming from RBF and kriging interpolations. And while NN interpolation still lags behind, this smoothing resulted in errors that are generally an order of magnitude smaller than those seen in Table 7, and of the same order of magnitude as the RBF and kriging methods. Once again the results from a non-smooth strain rate dataset are displayed in Figure 21.

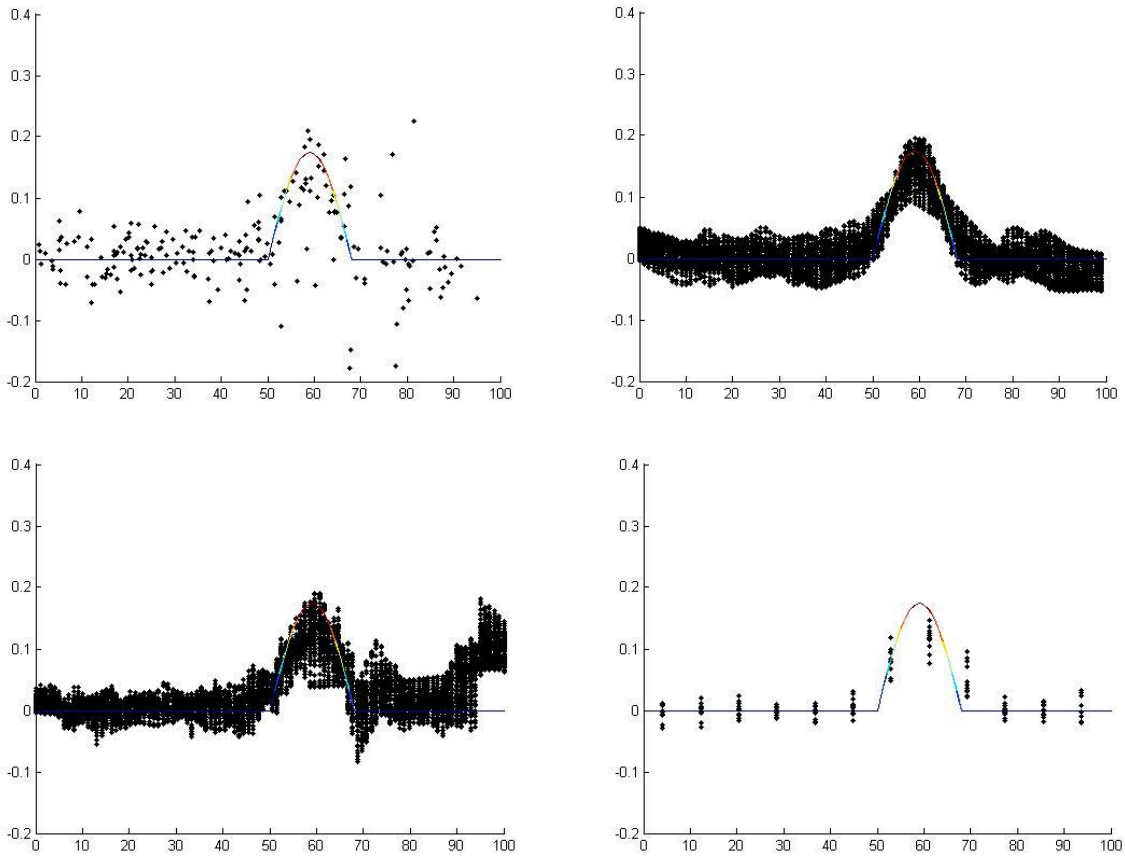

Figure 21. Strain rate estimates of the non-smooth strain rate flow with denoising, clockwise from top left a) natural neighbor interpolation, b) RBF, c) adaptive Gaussian window, d) kriging

It can be seen that the non-grid interpolation methods benefitted greatly from the denoising, and the underlying strain rate profile is better reflected in the results. The NN method still suffers from large outliers, even with an outlier removal scheme applied. Though it benefitted the most from the noise removal, NN interpolation consistently underperformed the RBF and kriging techniques. The final metrics of performance were computation time and ability to resolve small flow features. The results of time trials are presented in Figure 22. The number of data points in a 100 x 100 area was increased and the time to perform the interpolation recorded. Results are from tests performed on a Dell Precision PWS490 Intel® Xeon® CPU E5345 @2.33GHz with 16.00GB of RAM.

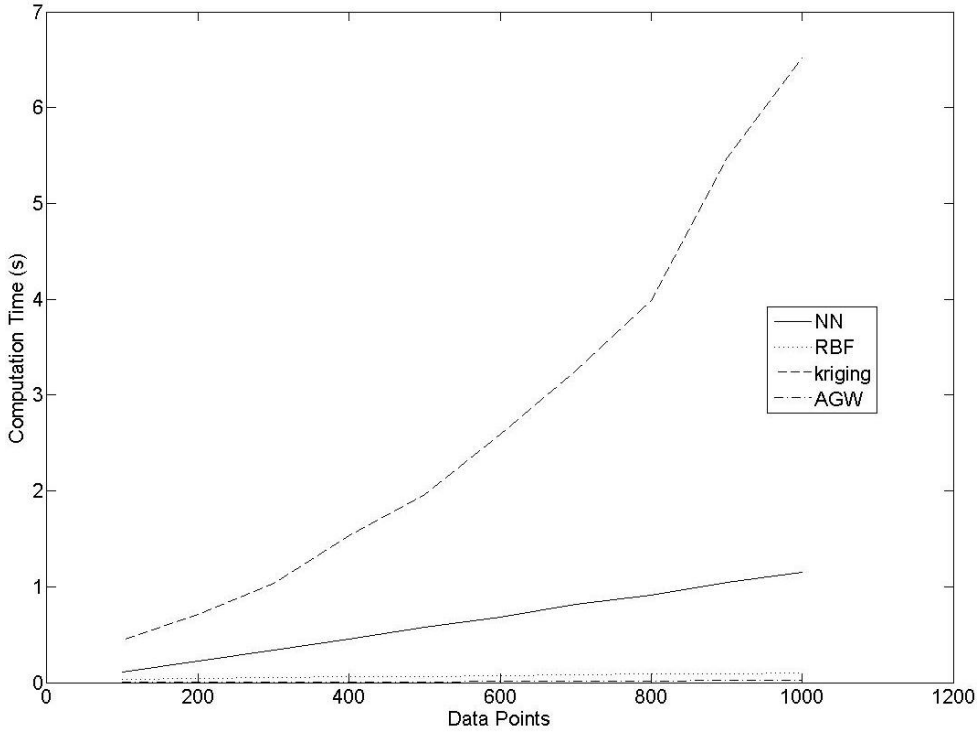

Figure 22. Computation time of various interpolation schemes

The fastest methods of interpolation from these results are RBF and AGW, with kriging being the slowest. In fact, above 1000 data points, the kriging method required more memory than was available to it on the experimental computer. The code which executes the NN interpolation was not specifically designed for efficiency, and is far less mature than the other techniques, and so could very conceivably be made to perform faster.

Finally, the interpolation methods are compared based on their ability to accurately resolve a small feature in a flow field. For this experiment, the non-smooth strain rate flow profile is used without noise and the parameter Δ is adjusted to control the feature size. The RMS error is calculated by examining the strain rate values within a region centered on the strain feature and three times

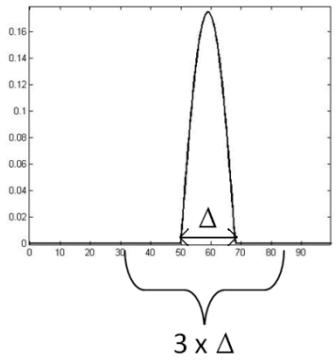

Figure 23 Interrogation area of error calculations.

its width (see Figure 23). Results are plotted in Figure 24 against the normalized flow feature size, $\Phi = \frac{\Delta}{mps}$, where *mps* is the mean particle spacing.

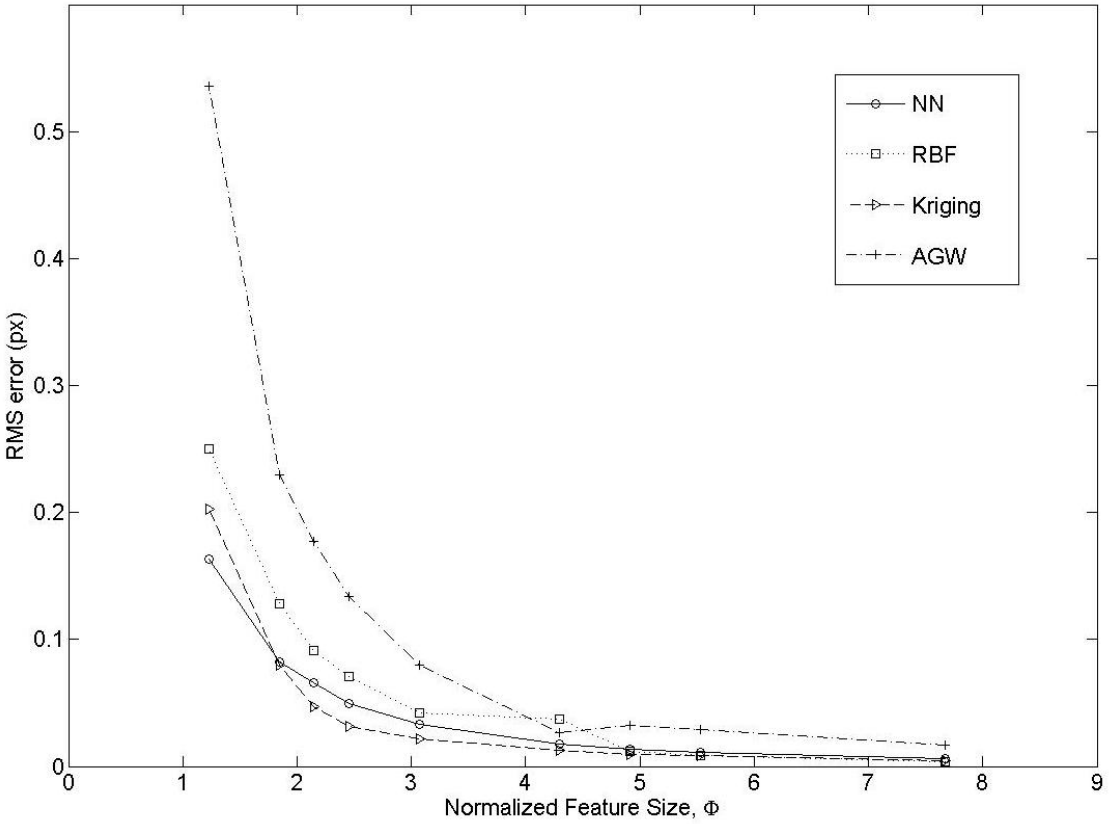

Figure 24. RMS error versus normalized feature size on a non-smooth strain rate flow profile.

It can be seen that features which are more than five times as large as the mean particle spacing can be accurately resolved by all four interpolation methods. As the normalized feature size is reduced to 4 and below, the grid based AGW method, which has interpolation nodes at 2.5 times the mean particle spacing, begins to result in higher errors. The non-grid based methods experience error divergence when the normalized feature size shrinks below three. The same trend appears in NN, RBF and kriging interpolation, and so it seems that the resolution of all three interpolation methods is primarily limited by the mean particle spacing.

6.4 Recommendations on Natural Neighbor Interpolation

Based on the results seen in section 6.3, it is clear that the current natural neighbor-based interpolation method is not competitive with existing interpolation methods when used to estimate gradients from scattered PTV data. The technique, though conceptually simple and computationally efficient, fails to outperform RBF or kriging interpolation on perfect PTV data. It is very sensitive to noise, and though it responds well to smoothing, it still fails to perform as well as RBF or kriging interpolation. Additionally, it does not even appear to have an advantage over RBF interpolation in terms of spatial resolution, as originally suggested (Duncan 2009). Some of this sub-par performance can be traced to the heart of the natural neighbor interpolation method. The function being interpolated (flow gradient) is unknown at each data point. In order to use the NN interpolation, an estimate of the gradient is made at each natural neighbor using something akin to a rudimentary backward differencing scheme (Equation 18), and so from the very beginning a great deal of error may be introduced into the NN interpolation, especially when the spacing between natural neighbors is large. Additionally, it could prove difficult to adapt a NN interpolation method for use in estimating gradients at the edge of a data set, such as finding shear stress on a wall, as the method performed poorly on these outer edges. Based on the results of this work, it is recommended that interpolation of PTV data be performed using either RBF or kriging interpolation, which have the added benefits of being well-established and widely used, or else some considerable variations be made to the NN method to make it more robust and accurate, such as directly finding the derivative of Equation 14 (Sukumar, 2003).

Chapter 7. CONCLUSION

In this thesis, a new particle tracking velocimetry technique was developed and methods for post processing of scattered velocity data were considered. While PIV results in a statistical

average of the local velocity in a flow, PTV can give the velocities of individual particles. Typically PTV has been used with low particle image density flows, since identifying individual particles becomes increasingly difficult when particle images overlap. By using a new cascade correlation method (Lei *et al*., 2012) to identify the centroids of overlapped particle images, PTV can be used with higher particle image density flows, and thus higher resolution velocity data can be obtained. A feature matching technique was adopted from computer vision which relied upon the principles of proximity and exclusion. This matching algorithm was guided with PIV results and combined with a cross-correlation term which took into account the principle of similarity when making matches. Particle matching was performed iteratively on overlapping interrogation windows and an outlier detection scheme was used to validate matches. This VB-PTV algorithm was tested on synthetic images of a moving wall flow and resulted in match yields of over 98% and reliability of matches of more than 99%. When the algorithm was tested on standard synthetic images from the Visualization Society of Japan, it identified more particles and made more correct matches than existing PTV methods. Tests performed on experimental images showed that this VB-PTV technique was suitable for real applications. Sensitivity tests suggested that, despite the ability to resolve overlapped particle images, this method still performs best at lower particle image densities and smaller particle image diameters. It was seen that match yield and reliability were degraded and the error of matches increased in highly straining flows. This trend was aggravated by high displacements.

In an effort to improve this performance, a simple modification to the proximity matrix used in the matching process was introduced. Essentially, the principle of proximity was relaxed and instead PIV was used to produce a more selective set of candidate particles in the second image frame which could be matched to a target particle in the first image frame. Two

modifications were proposed: one which relied only on an estimate of the distance to a correct match, and one which used both distance and directional estimates from PIV to guide matching. These modified matching methods reduced the error in high-displacement (maximum 25 pixels) high-shear flow, though they had less of an impact on shearing flow with small displacements (maximum 7 pixels). Though the results from reprocessing synthetic moving wall images showed little change from the original matching method, due in large part to the low gradients and small displacements in the flow, tests performed on a synthetic Oseen vortex flow and the standard VSJ images showed that the modifications could reduce errors by as much as 33% in images with unknown particle locations, and it improved the match yield and reliability of results from the VSJ images by 2 percentage points to over 99% in both cases—the best result when compared with other PTV methods. The difference between the performances of each of the two modifications was minor, but both showed improvements over the original matching method, and so their use was recommended in any future application of the VB-PTV technique.

Finally, consideration was given to post-processing of scattered PTV data. A method of obtaining accurate estimates of derivative properties, such as shear stress or vorticity, was desired. A natural neighbor-based method (NN) of derivative estimation was proposed which allowed for the direct calculation of flow derivatives at each PTV data point. This method was compared with a well-known grid-based adaptive Gaussian window interpolation method, as well as two methods which estimate a function which describes the velocity surface—radial basis function and kriging interpolation. Tests showed that the NN method consistently resulted in higher errors than RBF and kriging, and was highly sensitive to noise. Though it responded well to a denoising technique, NN interpolation remained prone to outliers and generally less accurate than other interpolation schemes. Additionally, the NN interpolation displayed little or

no improvement over the other interpolation methods when compared based on computational time and spatial resolution. RBF and kriging interpolation appear to be more useful methods for interpolation of scattered data, and they have the advantage of being widely used and well-developed.

A collaborative effort to develop a PTV algorithm for use at the University of Washington has been successful. This process has been shown to give accurate results in various synthetic and experimental applications, methods have been explored for extracting derivative information from the scattered data, and this tool is suitable for use in future particle imaging applications.

BIBLIOGRAPHY

1. Abrahamson S, Lonnes S. (1995). Uncertainty in calculating vorticity from 2D velocity fields using circulation and least-squares approaches. *Exp in Fluids.* 20: 10-20.
2. Adrian RJ, Westerweel J. (2011) Particle Image Velocimetry. Cambridge, MA: *Cambridge University Press.*
3. Adrian RJ, Yao CS. (1985). Pulsed laser technique application to liquid and gaseous flows and the scattering power of seed materials. *Applied Optics. Vol. 24* , 44-52.
4. Agui J, Jimenez J. (1987). On the performance of particle tracking. *Journal of Fluid Mech 185* , 447-468.
5. Angarita-Jaimes NC, Roca MG Towers CE, Read ND, Towers DP. (2009). Algorithms for the automated analysis of cellular dynamics within living fungal colonies. *Cytometry A 75(9)*, 768-780.
6. Baek S, Lee S. (1996). A new two-frame particle tracking algorithm using match probability. *Experimental Fluids 22(1)* , 23-32.
7. Belikov VV, Ivankov VD, Kontorovich VK, Koytnik SA, Semenov YuA. (1997). The non-sibsonian interpolation: a new mthod of interpolation of the values of a function on arbitrary set of points. *Computational Mathematics and Mathematical Physics*. 37(1): 9-15.
8. Bernardeau F, van de Weygaert R. (1996). A new method for accurate estimation of velocity field statistics. *Mon. Not. R. Astron. Soc.* 279: 693-711.
9. Brevis W, Niño Y, Jirka G H. (2011). Integrating cross-correlation and relaxation algorithms for particle tracking velocimetry. *Exp Fluids*, 50: 135-147.
10. Buhmann M, (2003). Radial Basis Functions. Cambridge, MA: *Cambridge University Press.*
11. Chirokov, Eric. (2006). Interpolation and approximation using Radial Base Functions (RBF). *A presentation included with a Matlab toolbox.* http://www.mathworks.com/matlabcentral/fileexchange/

12. Christ N, Friedberg R, Lee T. (1982). Random lattice field theory: General formulation. *Nuclear Physics* B202: 89-125.
13. Cowen E, Monismith S. (1997). A hybrid digital particle tracking velocimetry technique. *Experiments in Fluids 22(3)* , 199-211.
14. Dabiri, D. (2003). On the interaction of a vertical shear layer with a free surface. *J. Fluid Mech.* 480: 217-232
15. Desbrun M, Meyer M, Schröder P, Barr A. (2000). Anisotropic feature-preserving denoising of height fields and bivariate data. *Graphics Interface, 145-152.*
16. Desbrun M, Meyer M, Schröder P, Barr A. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. *Comp. Graphics, Vol .33, No. Annual Conference Series.*
17. Dong S, Meng H. (2000). Chebyshev spectral method and Chebyshev noise processing procedure for vorticity calculation in PIV post-processing. *Exp Therm and Fluid Sci.* 24: 47-59.
18. Duncan J. (2009). The development of advanced techniques for particle tracking velocimetry. *MS thesis, University of Washington, Department of Aeronautics and Astronautics.*
19. Duncan J, Dabiri D, Hove J, Gharib M. (2010). Universal outlier detection for particle image velocimetry (PIV) and particle tracking velocimetry (PTV) data. *Measurement Science and Technology 21(5)* , 057002.
20. Elghobashi S. (1994). On predicting particle-laden turbulent flows. *Applied Scientific Research 52(4)* , 309-329.
21. Foucaut J, Stanislas M. (2002). Some considerations on the accuracy and frequency response of some derivative filters applied to particle image velocimetry vector fields. *Meas. Sci. Technol.* 13: 1058-1071.
22. Fouras A, Soria J. (1998). Accuracy of out-of-plane vorticity measurements derived from in-plane velocity field data. *Exp in Fluids.* 25: 409-430.
23. Gharib M, Kremers D, Koochesfahani MM, Kemp M. (2002). Leonardo's Vision of Flow Visualization. *Experiments in Fluids 33(1)* , 219-223.
24. Gonzalez D, Cueto E, Martinez M, Doblare M. (2004). Numerical integration in natural neighbour Galerkin methods. *Int. J. Numer. Meth. Engng.* 60: 2077-2104.
25. Gunes Hasan, Sirisup S, Karniadakis GE. (2006). Gappy data: To Krig or not to Krig? *Journal of Computational Physics 212* , 358-382.
26. Hiyoshi H, Sugihara K. (1999). Two generalizations of an interpolant based on Voronoi diagrams. *Int. J. Shape Modeling.* 5(2): 219-231.
27. Journel A, Huijbregts CH. (1978) Mining Geostatistics. *Academic Press Inc. (London) LTD.*
28. Keane R, Adrian R. (1995). Super-resolution particle imaging velocimetry. *Measurement Science and Technology 6(6)* , 754-768.
29. Kim H, Lee S. (2002). Performance improvement of two-frame particle tracking velocimetry using a hybrid adaptive scheme. *Measurement Science and Technology 13(4)* , 573-82.
30. Lei Y-C, Tien W-H, Duncan J, Paul M, Ponchaut N, Mouton C, Dabiri D, Rösgen T, Hove J. (2012). A vision-based hybrid particle tracking velocimetry (PTV) technique using a

modified cascade-correlation peak-finding method. *Submitted for publication to Exp in Fluids* .

31. Liao Q, Cowen E. (2005). An efficient anti-aliasing spectral continuous window shifting technique for PIV. *Experiments in fluids 38(2)* , 197-208.
32. Little R, Rubin D. (2002). *Statistical Analysis with Missing Data.* New-York: Wiley-Interscience.
33. Lophaven NS, Nielsen H, Sondergaard J. (2002). DACE Matlab Kriging Toolbox Version 2.0. *Informatics and Mathematical Modelling* .
34. Luff J, Drouillard T, Rompage A, Linne M, Hertzberg J. (1999). Experimental uncertainties associated with particle image velocimetry (PIV) based vorticity algorithms. *Exp in Fluids.* 26: 36-54.
35. Luo B, Hancock E. (2002). Iterative Procrustes alignment with the EM algorithm. *Image and Vision Computing 20* , 367-369.
36. Marxen M, Sullivan P, Loewen M, Jähne B. (2000). Comparison of Gaussian particle center estimators and the achievable measurement density for particle tracking velocimetry. *Experiments in Fluids 29(2)* , 145-153.
37. Mikheev AV, Zubtsov V. (2008). Enhanced particle-tracking velocimetry (EPTV) with a combined two-component pair-matching algorithm. *Measurement Science and Technology 19(8)* , 085401.
38. Nogueira J, Lecuona A, Rodriguez P, Alfaro J, Acosta A. (2005). Limits of the resolution of correlation PIV iterative methods. *Experiments in Fluids 39(2)* , 305-313.
39. Nogueira J, Lecuona A, Rodriguez A. (2001). Identification of a new source of peak locking, analysis and its removal in conventional and super-resolution PIV techniques. *Experiments in fluids 30(3)* , 309-316.
40. Nogueira J, Lecuona A, Rodríguez P A. (2001). Local field correction PIV, implemented by means of simple algorithms, and multigrid versions. *Measurement Science and Technology 12(11)* , 1911-1921.
41. Ohmi K, Li H Y. (2000). Particle-tracking velocimetry with new algorithms. *Measurement Science and Technology 11(6)* , 603-616.
42. Okamoto, K., Nishio, S., Saga, T. and Kobayashi, T. (2000). Standard images for particle-image velocimetry. *Measurement Science and Technology 11(6)* , 685-691.
43. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber. 9* , 62-66.
44. Panday S P, Ohmi K, Nose K. (2011). An ant colony optimization based stereoscopic particle pairing algorithm for three-dimensional particle tracking velocimetry. *Flow Measurement and Instrumentation*, 22: 86-95.
45. Pilu, M. (1997). Uncalibrated stereo correspondence by singular value decomposition. *IEEE Computer Vision & Pattern Recognition Conference, 1997,* (pp. 261-266). San Juan , Puerto Rico.

46. Ponchaut N, Mouton C. (2005). 3D Particle Tracking Velocimetry Method: Advance and Error Analysis. *GALCIT Report FM2005.004.*
47. Ponchaut, N. (2005). Part I: 3DPTV: Advances and Error Analysis; Part II: Extension of Guderley's Solution for Converging Shock Waves. *PhD Thesis, California Institute of Technology.*
48. Raffel M, Willert C, Kompenhans J. (1998). Particle Image Velocimetry: A Practical Guide. *Berlin: Springer*.
49. Ruhnau P, Guetter C, Putze T, Schnorr C. (2005). A variational approach for particle tracking velocimetry. *Measurement Science and Technology 16(7)* , 1449-1458.
50. Saarenrinne P, Piirto M. (2000). Turbulent kinetic energy dissipation rate estimation from PIV velocity vector fields. *Exp in Fluids [Suppl.]* S300-S307
51. Sacks J, Welch W, Mitchell T, Wynn H. (1989). Design and analysis of computer experiments. *Statistical Science.* Vol 4. No. 4, 409-435.
52. Saga T, Kobayashi T, Segawa S, Hu H. (2003). Development and evaluation of an improved correlation based PTV method. *6th International Symposium on Fluid COntrol, Measurement and Visualization.* Sherbrooke Canada.
53. Scarano, F. (2003). Theory of non-isotropic spatial resolution in PIV. *Experiments in Fluids 35(3)* , 268-277.
54. Schaap W, van de Weygaert R. (2000). Continuous fields and discrete samples: reconstructions through Delaunay tessellations. *Astron. Astrophys.* 363: L29-L32.
55. Schindler L, Moroni M, Cenedese A. (2010). Spatial-temporal improvements of a two-frame particle-tracking algorithm. *Meas Sci Technol.* 21: 115401
56. Schonemann, P. (1966). A generalized solution of the orthogonal Procrustes problem. *Psychometrika 31* , 1-10.
57. Scott G, Longuet-Higgins. H. (1991). An algorithm for associating the features of two images. *Biological Sciences 244* , 21-26.
58. Sibson R. (1980). A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society.* 87: 151-155.
59. Song X, Yamamoto F, Iguchi M, Murai Y. (1999). A new tracking algorithm and removal of spurious vectors using Delaunay Tesselation. *Experiments in Fluids 26(4)* , 371-380.
60. Spedding G, Rignot E. (1993). Performance analysis and application of grid interpolation techniques for fluid flows. *Exp. In Fluids 15,417-430.*
61. Stellmacher M, Obermeyer K. (2000). A new particle tracking algorithm based on deterministic annealing and alternative distance measures. *Experiments in Fluids 28(6)* , 506-518.
62. Sukumar N. (2001). Sibson and non-Sibsonian interpolants for elliptic partial differential equations. *First MIT Conference on Computational Fluid and Solid Mechanics.*
63. Sukumar N, Bolander J E. (2003). Numerical computation of discrete differential operators on non-uniform grids. *CMES* vol. x, no. x, pp.1-15.

64. Takehara K, Adrian R, Etoh G, Christensen K. (2000). A Kalman tracker for super-resolution PIV. *Experiments in Fluids 29(7)* , s034-s041.
65. Tanaka T, Eaton J. (2007). A Correction method for measuring turbulence kinetic energy dissipation rate by PIV. *Exp Fluids.* 42: 893-902.
66. Uemura T, Kawahara G, Yamamoto F. (1989). A high speed algorithm of image analysis for real time measurement of two-dimensional velocity distribution. *Flow Visualization* , 129-34.
67. Ullman, S. (1979). *The Interpretation of Visual Motion.* Cambridge MA: MIT Press.