

© Copyright 2021

Yi-Lin Liu

Particle Identification Technique for Particle Tracking Velocimetry Application in Noiseless Image

Yi-Lin Liu

A dissertation

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2021

Reading Committee:

Dana Dabiri, Chair

James Hermanson

Program Authorized to Offer Degree:

Aeronautics & Astronautics

University of Washington

Abstract

Particle Identification Technique for Particle Tracking Velocimetry Application in Noiseless Image

Yi-Lin Liu

Chair of the Supervisory Committee:
Professor Dana Dabiri
Aeronautics & Astronautics

This thesis will investigate particle identification in the 2D noiseless image. The modified cascade cross-correlation method (MCCM) is first introduced. The idea of it is to perform cross-correlation on images and apply a nonlinear least-square solver to give sub-pixel accuracy. Next is the support set method, which uses a morphological way to segment images. An iterative scheme is applied to each segment to acquire particle location. Finally, the technique developed from 3D tomographic PIV which is projected to 2D then utilizes nonnegative least-square (NNLS) to retrieve particle location. This method is superior in the ability to quantitatively find particles and the accuracy in constant particle size images. For images that the particle size varied, we proposed a method that required MCCM to determine the size to apply in NNLS and the result from NNLS to reconstruction image. After getting the reconstructed image, the residual can be calculated. The particle position corresponded to the lowest residual is selected to be the result that is shown better than other methods.

Table of Contents

SECTION 1. INTRODUCTION	10
SECTION 2. SYNTHETIC IMAGE AND PERFORMANCE METRICS.....	12
2.1 Synthetic images	12
2.2 Overlap ratio.....	13
2.3 Performance metrics.....	13
SECTION 3. METHODOLOGY FOR FINDING PARTICLE PEAK LOCATION	14
3.1 Modified cascade cross-correlation method.....	14
3.1.1 Optimized residual method	17
3.2 Support Set Method.....	19
3.3 Particle reconstruction and Nonnegative least-square method	23
3.3.1 The PSF model	24
3.4 MCCM-NNLS.....	27
SECTION 4. NUMERICAL TESTS RESULTS.....	28
4.1 Two-particle tests	28
4.2 Constant radius particle tests.....	30
4.3 Various radius particle tests	34
4.3.1 Impact of inaccurate σ_{PSF}	34
4.3.2 The effect of ΔE	38
SECTION 5. CONCLUSION AND FUTURE WORK	41

BIBLIOGRAPHY	42
--------------------	----

List of Figures

Figure 1. (a) $r = 0.6$ at density = 0.01 (b) $r = 0.6$ at density = 0.1.....	12
Figure 2. Two $r = 1$ particles at <i>overlap ratio</i> = 60%.....	13
Figure 3. Rosin's thresholding visual illustration[10]	16
Figure 4. Comparison of Rosin's method and Otsu's method, the red square illustrates that after Otsu's thresholding only 2 pixels is left in the blob which may cause the information loss and lead to the inaccurate fitting result	16
Figure 5. The particle identification algorithm's process flow proposed in [5].....	17
Figure 6. The process flow for optimized residual method	18
Figure 7. The ratio of real particle number N_{answer} and the number found by MCCM N_{found} in different N_{ppp} condition	19
Figure 8. The EDT steps (a) The original image (b) Eroded image after 34 steps (c) The final result of erosion, where the illuminated pixel are the local peaks found by eroding (d) The segmentation achieved by dilation from the local peaks.	21
Figure 9. The process flow for single image support set method	22
Figure 10. Support set blob in a noisy image. The red circle shows one of many support set blob that doesn't contain particle in it.....	22
Figure 11. Point light source convoluted with PSF visual demonstration.....	23
Figure 12. The EDT visual illustration (a) The original image (b) The image from NNLS process (c) Dilation image using square structure element, complete by <i>imdilate</i> and <i>strel</i> in MATLAB (d) The local maximum found from each dilated area from(c))	26
Figure 13. Process flow of MCCM-NNLS method on various radius particle image.....	28

Figure 14. Discretized intensity profile of 2 identical particles($r = 0.6$) with overlap ratio = 40% at different sub-pixel center location	29
Figure 15. Comparison between NNLS and MCCM in 2 particle overlapping test at $r =$ $0.6, 1, 1.2$	30
Figure 16. The yield and reliability performance in different particle density conditions between NNLS, MCCM, MCCM + optimized residual and Support Set method at $r = 0.6$	31
Figure 17. The NNLS method computational time divided by MCCM computational time of different ΔE under different N_{ppp} . The hardware spec is Dell XPS15 Intel® CPU i7-7700 @ 2.80GHz with 16GB RAM	32
Figure 18. The performance of the proposed technique MCCM-NNLS on $r = 0.6$ image.....	33
Figure 19. Yield and reliability result when applying different σ_{PSF} on 30 images that $r = 0.6$, $N_{ppp} = 0.1$	35
Figure 20. The qualitative view of the impact of underestimating the real size of particle while using NNLS($r = 0.6, \sigma_{PSF} = 0.4$). The red circles are the found particle location and most of them are False Positive.	35
Figure 21. Result of taking the median of found radius from MCCM to be σ_{PSF} in $r = 0.6 -$ 1.4 image.....	36
Figure 22. Qualitative view of $r = 0.6$ and $r = 0.8$ particles at (a) <i>overlap ratio</i> = 20% applying $\sigma_{PSF} = 0.6$ (b) <i>overlap ratio</i> = 70% applying $\sigma_{PSF} = 1$, the blue circles are the found particle location by NNLS	37
Figure 23. The normal distribution fit of real particle size vs particle size detected by MCCM at $r = 0.6 - 1.4$	38

Figure 24. Qualitative view of dispersion of False Positive. $r = 1$, $\sigma_{PSF} = 0.6$, (a) $\Delta_E = 15$ (b)	
$\Delta_E = 120$	39
Figure 25. Performance comparison between MCCM + optimized residual and MCCM-NNLS at	
$r = 1 - 1.2$ image	40
Figure 26. Performance comparison between MCCM + optimized residual and MCCM-NNLS at	
$r = 0.6 - 1.4$ image	40
Figure 27. Error comparison between MCCM-NNLS with different and MCCM + optimized	
residual at (a) $r = 1 - 1.2$, (b) $r = 0.6 - 1.4$	41

List of Tables

Table 1 20

Table 2 27

SECTION 1. INTRODUCTION

Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) are the techniques used in experimental fluid mechanics to investigate flow velocity fields. Seeding particles implanted in a flow field are visible within a narrow laser sheet and imaged in successive frames separated by a tiny time delay. For PIV, it investigates flow behavior in a controlled region over time. On the other hand, PTV tracks individual particles in the flow. Identification of particle position and size is critical in PTV. Before the mapping procedure on 3D, particles would be identified in a 2D image. The inaccuracy of particle detection can lead to erroneous displacement results. For example, a particle travels 4 pixels in between two timeframes, but the detection process has an offset of 1 pixel, then the error of PTV is up to 25%. Moreover, the particle density in the imaging region should be as high as possible to effectively use the PTV approach's high spatial resolution characteristic. As a result, while determining particle positions in those dense images, overlapping particles become a serious concern. Therefore, in 3D PTV technology, the precision and qualitative capacity to determine where the particles are in a plane image play a large role.

Several methods have been developed trying to detect particles in 2D PTV images. A typical way to resolve such problem is to apply Gaussian line or surface fit to find the peak with subpixel accuracy from finding local intensity maximum, such as [1]. [2] developed a better 4-point Gaussian estimator that accounts for pixel discretization effects using integral formulations, reducing the inaccuracy of standard 3-point Gaussian fit. Other approaches include using “mask” image to perform cross-correlation and find the peak. The particle correlation mask method (PMCM) was introduced in [3], which shows how to find identical particles by using a fixed size of the particle mask. For those particles, the normalized critical distance to independently

identify two particles is roughly unity, but it increases as the intensity difference grows. PMCM can resolve overlapping particles up to a minimum critical separation distance, however, if the particles' peaks are close enough to merge to a single peak, PMCM cannot detect it. Cascade cross-correlation method (CCM) is proposed by [4] and shows that a cascade of cross-correlation, rather than applying cross-correlation once with the particle mask images, reduces the correlation peak, and make the critical separation distance lower that two overlapping particles can be resolved. [5] modified the CCM procedure that after the cascade cross-correlation process, nonlinear least-square fitting is applied to get better accuracy of particles' center.

Aside from directly obtaining particle location from a 2D image, some methods such as the iterative particle reconstruction (IPR) is introduced in [6], the advanced IPR can also be seen in [7], use 3D triangulation technique to deal with high-density images. 2D detection is only a part of the whole scheme. It finds particles with the help of multiple images, which is the reason why this kind of process can have a great ability to find peak position. However, the goal for this thesis is to identify particles solely by looking at a single image because sometimes the laboratory may have physic constraints that cannot set multiple cameras. In addition, the accuracy of 2D detection may also improve the IPR algorithm.

This thesis is structured as follows. Section 2 will explain how synthetic images are generated by 2D Gaussian distribution, the tests conducted including overlapping particle tests, and the term used throughout the thesis. In Section 3, 3 techniques will be discussed to identify the location of particles. The numerical results of these methodologies are presented in Section 4.

SECTION 2. SYNTHETIC IMAGE AND PERFORMANCE METRICS

2.1 Synthetic images

Synthetic particle images will be used in the rest of the thesis. The particles are perfectly 2D Gaussian-shaped, which means there is no skewness in the images which means that every particle is symmetric. The i th particle in the image $I(j, k)$, which is represented by the equation (1), has four variables - x and y location(j, k), intensity I_0 and radius r (and diameter $D \approx 4 \times r$).

$$I(j, k) = \sum_{i=1}^N (I_0)_i e^{-\frac{x_j(j,k)^2}{2r_i^2}} \quad (1)$$

The diameter determines how many pixels. For an image, variables are the width, length, and particle density, N_{ppp} which stands for particle per pixel. The example image of different N_{ppp} is shown as follow:

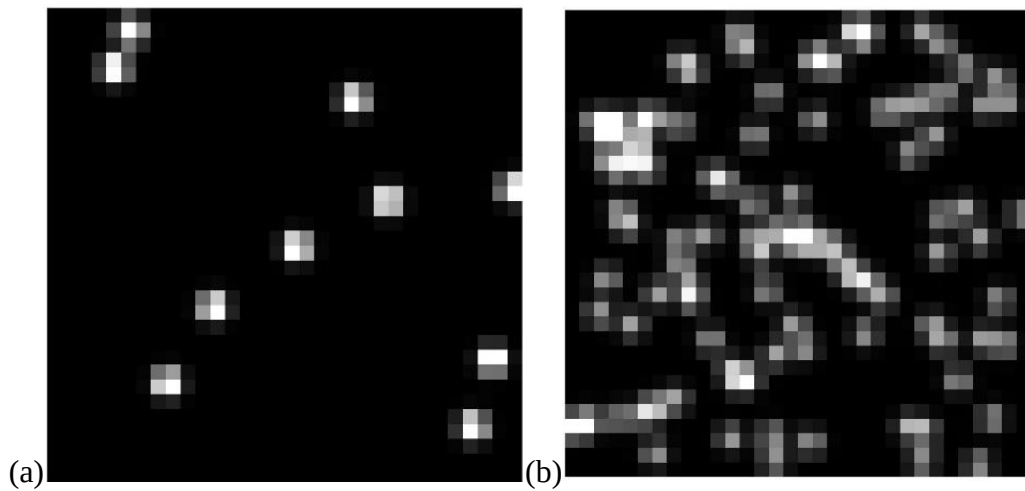


Figure 1. (a) $r = 0.6$ at density = 0.01 (b) $r = 0.6$ at density = 0.1

2.2 Overlap ratio

The metric that will be used in the tests later for examining how well the algorithm deal with the overlapping is the overlap ratio, which is proposed in [5].

$$Overlap\ ratio = \frac{D - L}{D} \quad (2)$$

where D is the diameter of the particle and L is the center distance between 2 particles. For this thesis, two identical particles will be used in the tests.

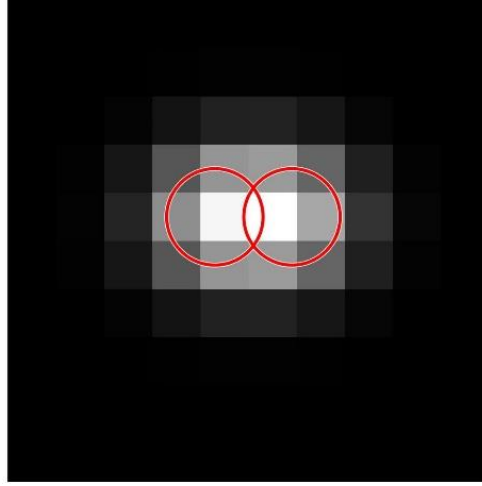


Figure 2. Two $r = 1$ particles at *overlap ratio* = 60%

2.3 Performance metrics

The quantitative metric for algorithm's performance and robustness are suggested by [8], yield and reliability.

$$Yield = \frac{\#TP}{\#TP + \#FN} \quad Reliability = \frac{\#TP}{\#TP + \#FP} \quad (3)$$

TP stands for true positive. A detected particle is considered as true positive if it is closest one to a real particle location and it is in the error tolerance, which is set to 0.5 pixels in this thesis.

False Positive(FP) particle means the detection is not close enough to any answer location or it is in the neighborhood but is not the nearest one. Lastly, False Negative(FN) is an answer particle that has no particle inside its neighborhood. As a result, the yield is the proportion of genuine detections to the total number of particles created, and the reliability is the proportion of true detections to all detections. With the computation of the mean localization error, the accuracy of the detections considered as TP is further quantified as:

$$Error = \frac{1}{\#TP} \sum_{i=1}^{N_{TP}} \|x_{TP_i} - x_{P_i}\| \quad (4)$$

where P_i is the i th particle, TP_i is the detection pair corresponding to P_i and x is their location in the image. This metric indicates the real distance between the TP position and the precise location of the particle it is linked with. Noted that this type of measurement procedure can only be used on synthetic PTV images because these two performance trials required the real particle center position for the image.

SECTION 3. METHODOLOGY FOR FINDING PARTICLE PEAK LOCATION

3.1 Modified cascade cross-correlation method

The modified cascade cross-correlation method(MCCM) from [5] avoided the peak-lock issue from the previous CCM method proposed by [4]. MCCM brought the estimated peak location found by cascade cross-correlation process into a nonlinear least-square fitting solver to get a

continuous location instead of pixel-by-pixel peak. This process can be done by MATLAB command *lsqcurvefit*. The least-square optimization problem is to minimize $\mathcal{X}^2$ in equation.

$$\mathcal{X}^2 = \sum_{i=1, j=1}^{j=m, k=n} \left[I(j, k) - \sum_{i=1}^N (I_0)_i e^{\frac{x_i(j, k)^2}{2r_i^2}} \right] \quad (5)$$

where the (x_i, y_i) is particle location, $(I_0)_i$ is the intensity and r_i is the intensity for the i th blob.

Prior to the fitting process, raw particle image is segmented by Otsu method[9], *graythresh* command in MATLAB, is used to roughly filter the background noise and segment “blobs”.

However, the intensity distribution of a clean image is unimodal. In such a case, the Otsu thresholding approach may cause the blob to lose its Gaussian shape characteristic, as illustrated in Figure 4, resulting in inaccuracy when doing nonlinear least-squares fitting. As a result,

Rosin's thresholding[10] is presented as a unimodal thresholding procedure. Rosin's method can preserve more Gaussian distribution traits as shown in Figure 4 which benefits the fitting step.

Each blob will be performed cross-correlation several times, the mask radius is reduced by 1 pixel each time until it reaches the user-determined minimum. The process flow block diagram of this technique is shown in Figure 5. The reasons for this step are (1) to separate heavily overlapped particles furthermore and (2) to improve the ability detecting wide range particle size.

The cross-correlation process in MATLAB is done by command *xcorr2* inputting blob and mask image to get the pixel that has the highest correlation the 2D Gaussian distribution.

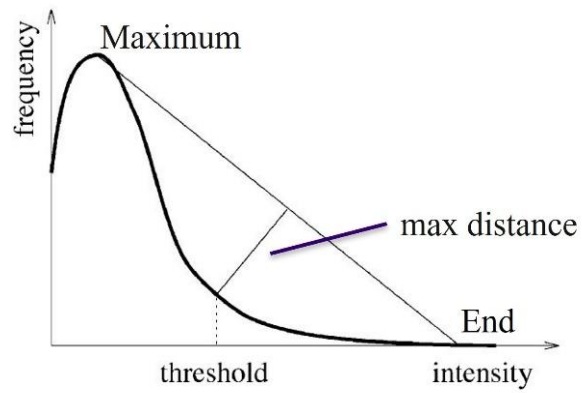


Figure 3. Rosin's thresholding visual illustration[10]

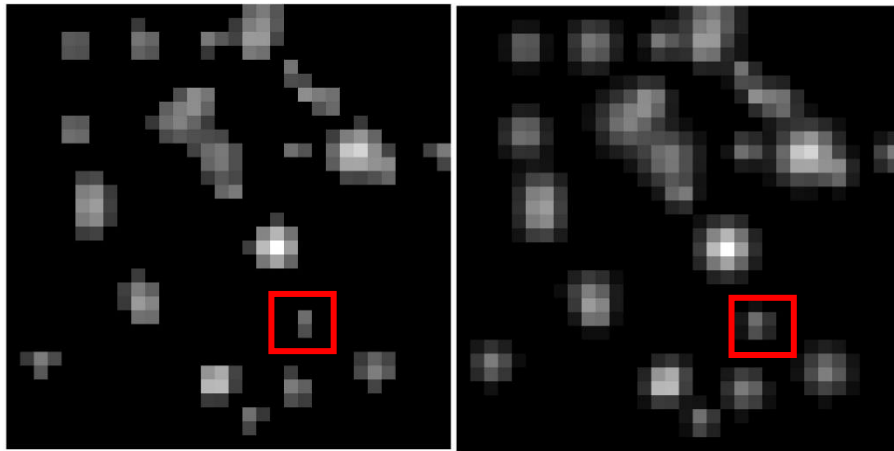


Figure 4. Comparison of Rosin's method and Otsu's method, the red square illustrates that after Otsu's thresholding only 2 pixels is left in the blob which may cause the information loss and lead to the inaccurate fitting result

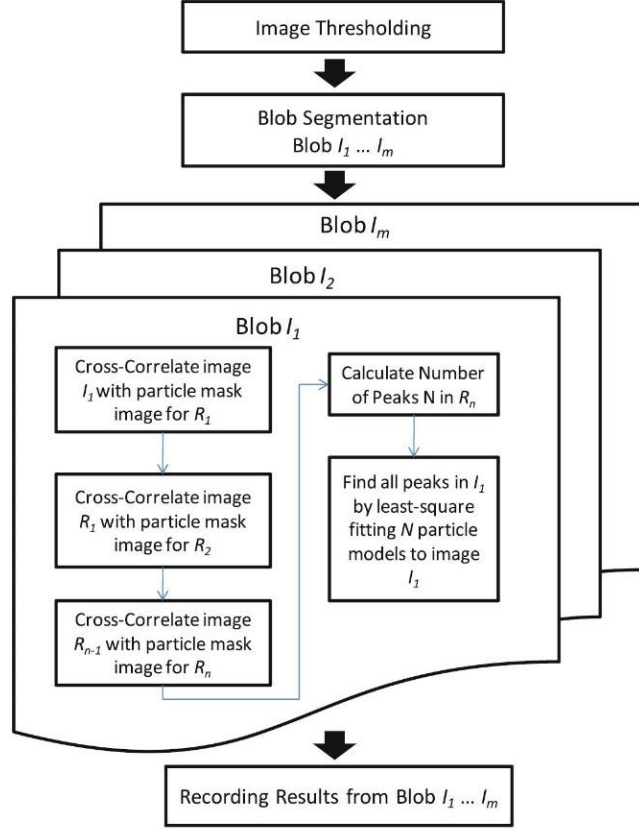


Figure 5. The particle identification algorithm's process flow proposed in [5]

3.1.1 Optimized residual method

As seen above, the performance of MCCM highly depends on the number of particles found N in the cross-correlation process. To improve the method, we suggested a process based on the difference between the reconstructed particle image and the original image, that is, the residual of the image. Residual is set to be the sum of intensity difference divided by the number of blob image pixels:

$$residual = \frac{\sum_{j=1, k=1}^{j=m, k=n} |I(j, k) - I_{reconstruct}(j, k)|}{mn} \quad (6)$$

This kind of process can be seen in [11]. Since the particle number N is usually underestimated, as shown in Figure 7, due to the limitation of cross-correlation can only have the initial particle location on integer pixel in the high-density image. Once we have the position, size, and intensity information from MCCM, we can begin residual optimization and acquire the difference image. Secondly, we add the pixel that has the largest intensity difference to the initial value for next non-linear least-squares fitting process. If residual of the least-squares fit from new initial value drops, we continue the procedure until the residual stop decreasing. In Section 4, we can observe that the reliability is down at low N_{ppp} but the yield increased about 10% in high density condition.

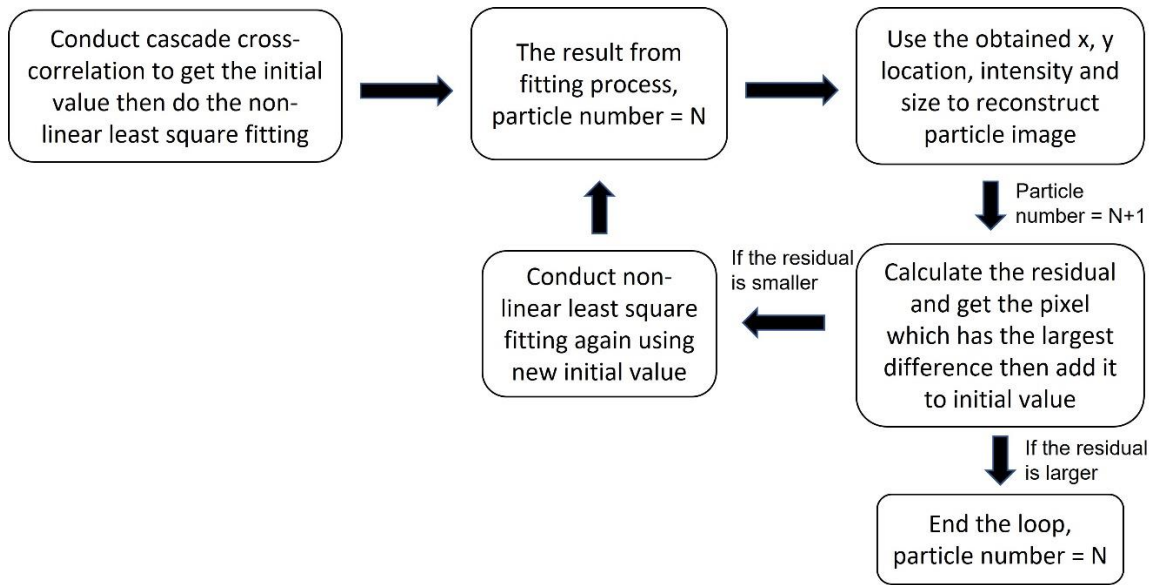


Figure 6. The process flow for optimized residual method

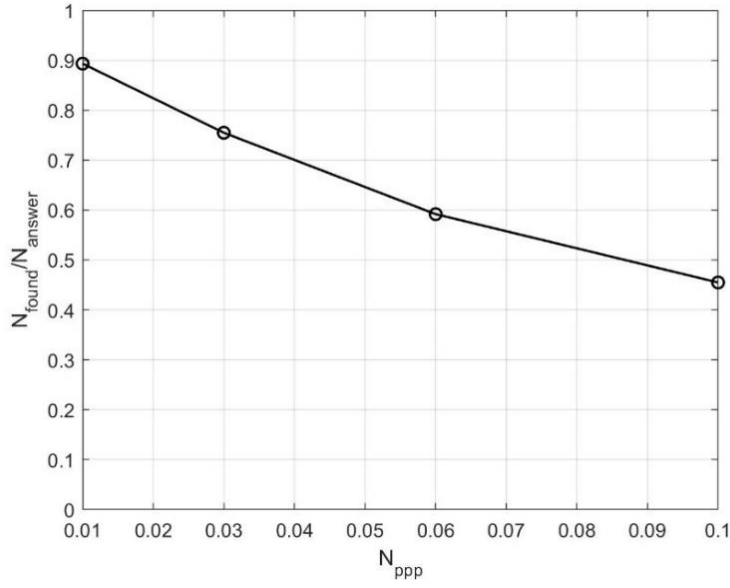


Figure 7. The ratio of real particle number N_{answer} and the number found by MCCM N_{found} in different N_{ppp} condition

3.2 Support Set Method

The support set method is suggested by [12]. It is based on a morphological way – Erosion Dilation Thresholding (EDT). To form the support set, the raw image will be eroded first. This erosion step can preserve the local peak of the image. Once we got the peak location, the dilation step would be performed and finally, the segmentation of the image is complete, as shown in Figure 8. The EDT details were discussed in [13]. In the original paper, they did not try to locate where the particles are using a single PTV image. Instead, they only test the consistency and robustness by conducting 2 cases of initialization of location, shape, and intensity. Case 1 is giving the perfect initial value to the continuous least-squares solver, and the second case is only providing the exact number of particles in a support set. The result for these two cases:

	$N_{ppp} = 0.1$
Case 1	98.5%
Case 2	91.5%

Table 1

However, the thing we want to achieve in this thesis is to resolve the particle location in a single noise-free image, so we did not implement the 3D particle and 2D particle tracking technique discussed in [12] to find multiple peak locations. Instead, we tried to use the same least-squares method to fit each segmented image. Every support set is treated as a “blob” like in MCCM and the idea of finding optimal particle location is similar to the residual method mentioned previously. Every “blob” starts with $N = 1$ at the peak location found in the erosion step. Then we try to squeeze an additional particle into the support set until the calculated residual is higher than previous one. The whole process flow can be seen in Figure 9.

The support set method can better separate images into blobs than the thresholding process used in MCCM and still preserve the Gaussian particle feature. The less particle contains in a single blob, the higher chance the residual method to find the correct particle number. However, the critical drawback of this morphological process is that EDT step will have difficulties applying to a noisy image. Peak preserving trait of image erosion step would not distinguish the peak in the noise region from real particle peak which will increase False Positive lowering the reliability. Since the target for the next step of particle identification is to take noise into consideration (Figure 10), and unlike MCCM and later mentioned NNLS have already discussed noisy conditions in the original paper, the support set method will not be further investigated in some tests later in this thesis.

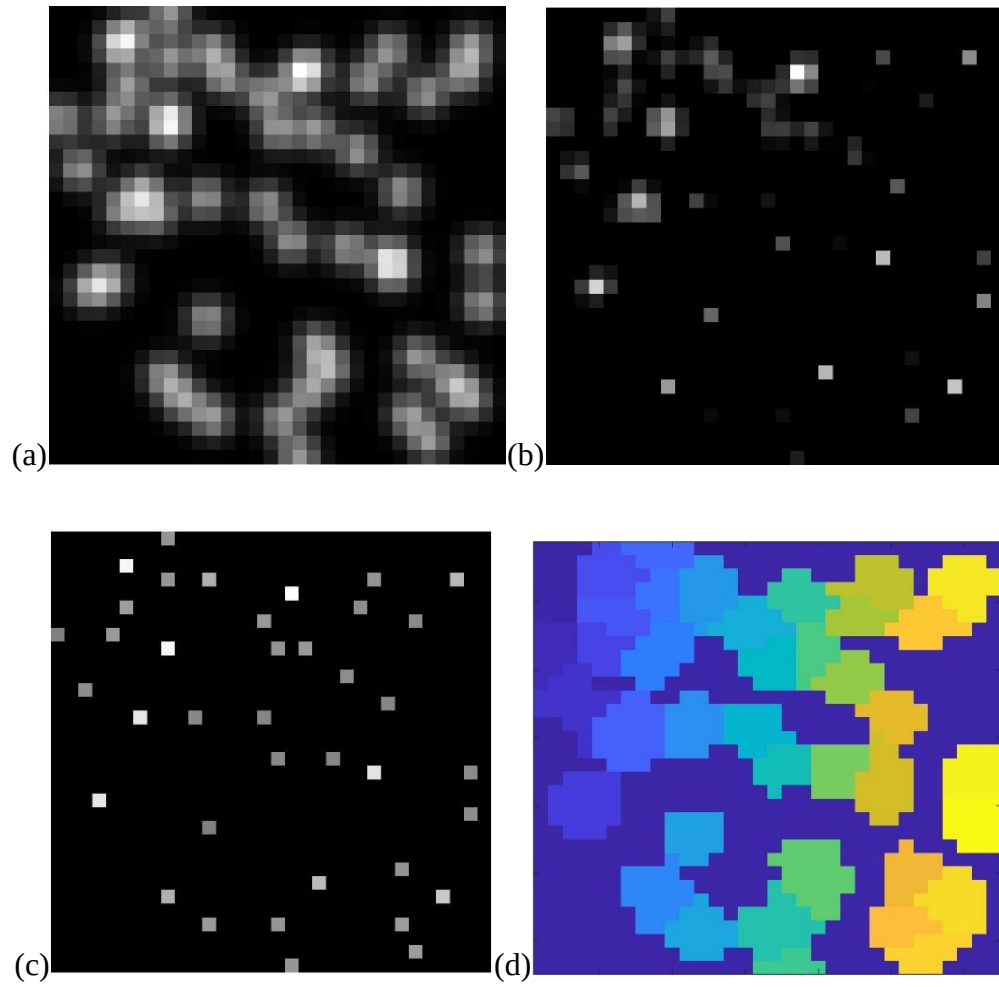


Figure 8. The EDT steps (a) The original image (b) Eroded image after 34 steps (c) The final result of erosion, where the illuminated pixel are the local peaks found by eroding (d) The segmentation achieved by dilation from the local peaks.

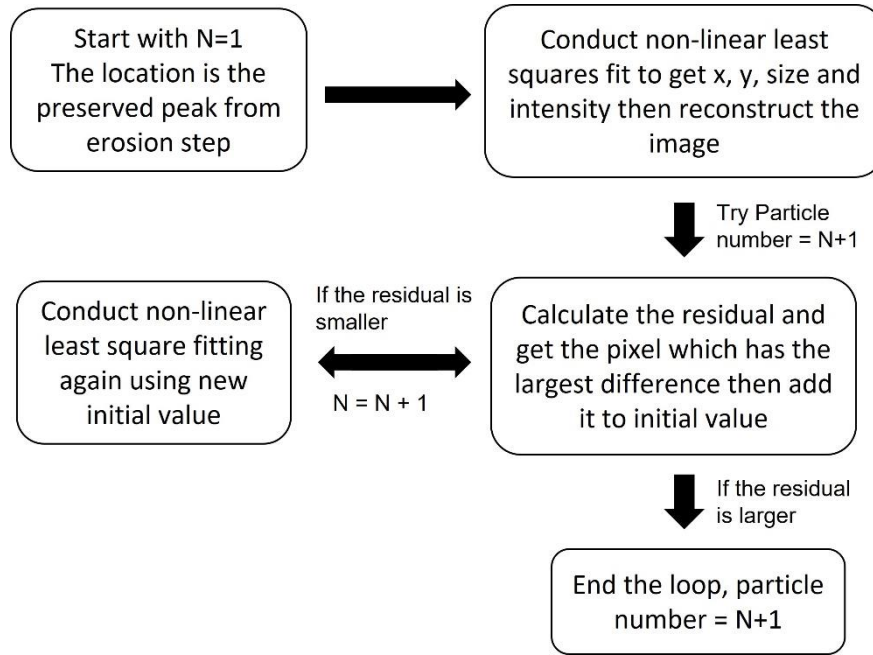


Figure 9. The process flow for single image support set method

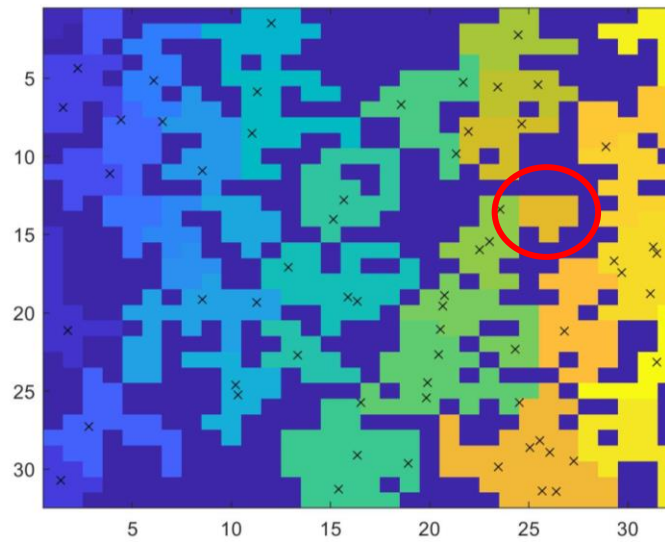


Figure 10. Support set blob in a noisy image. The red circle shows one of many support set blob that doesn't contain particle in it.

3.3 Particle reconstruction and Nonnegative least-square method

The classic tomographic PIV technique introduced in [14] is developed for robustly and accurately reconstructing 3D flow field. The mathematical concept behind this method is the projection model of particles in 3D space and corresponding 2D space multi-view images.

$$Ax = b \quad (7)$$

where b collects pixels from multi-view images, x collects particle intensities within the lit volume, and A is a matrix that depends on the projection's type.

The point spread function (PSF) describes the response of an imaging system to a point source or point object. The particle image can be seen as a light point source convoluted by PSF as shown in Figure 11.

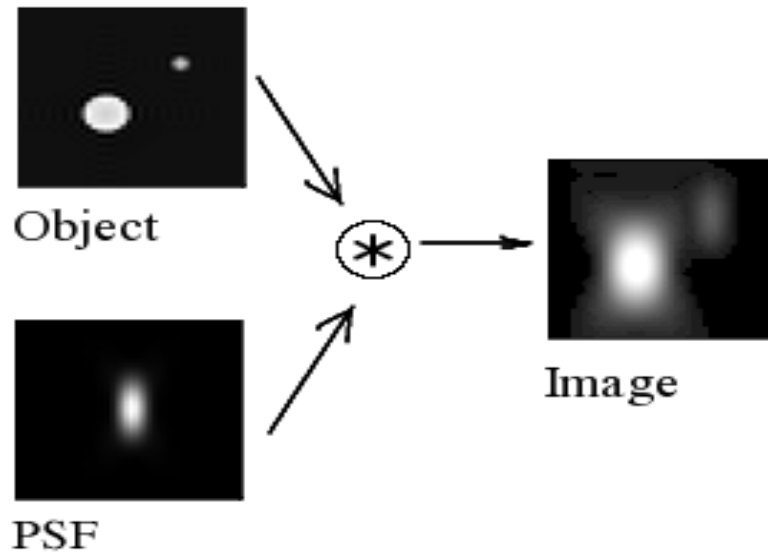


Figure 11. Point light source convoluted with PSF visual demonstration

The PSF is a feature of the camera system, not of the source and could lead to accuracy losses in the estimation.

3.3.1 The PSF model

The image model presented in [15] is shown in (8). Any light source point in 3D space located at X has a geometrical image in the focus plane of a particular camera located at $F(X)$, and the source intensity E is used shared between the afflicted pixel and its impacted neighbor pixels according to the weights assigned by PSF h_x . The image I at point $x \in R^2$ can be modelled as:

$$I(x) = \sum_{p=1}^P E_p h_{x_p} (x - F(\mathbf{X}_p)) \quad (8)$$

where P is the number of particles, E_p is the source intensity field of P particles in the image, X_p is the location of these particles, $F: R^3 \rightarrow R^2$ is the projection operator for a 3D point from volume to plane and h is the PSF. Here, the PSF is written as 2D Gaussian intensity distribution where the only variable is its standard deviation σ , which also can be treated as the radius of particles r .

In [15], the equation was modified to 2D sense. The intensity $I(x)$ is discretized by camera and can be transferred into an array of N pixels, each denoted $\mathbf{k} = (k_1, k_2)$. The discretized 2D version of equation :

$$I(k) \approx \sum_n h(k - n\Delta_E) E(n) \quad (9)$$

where I is the reshaped vector of the image and indexed by k , E is a vector that contains point source information in sub-pixel space. The M sub-pixel space is indexed by n of width Δ_E . The selection of Δ_E is discussed in [16] in the tradeoff between computing time and precision. The equation can be rewritten to the form:

$$I = WE \quad (10)$$

where $W = h(k - n\Delta_E)$ is called weight matrix indexed by k in pixel space and n in sub-pixel space. The PSF is

$$h(x, y) = \frac{1}{4} \left(\operatorname{erf} \left(\frac{x + 0.5}{\sqrt{2}\sigma_{PSF}} \right) - \operatorname{erf} \left(\frac{x - 0.5}{\sqrt{2}\sigma_{PSF}} \right) \right) \times \left(\operatorname{erf} \left(\frac{y + 0.5}{\sqrt{2}\sigma_{PSF}} \right) - \operatorname{erf} \left(\frac{y - 0.5}{\sqrt{2}\sigma_{PSF}} \right) \right) \quad (11)$$

where erf is the error function and σ_{PSF} is standard deviation and the radius of particles.

Now what we need to do is inversely retrieve the intensity field E . Since the weight matrix W is short and fat that has much more columns than rows, it is ill-posed and theoretically has an infinite number of solutions. Therefore, the key is to figure out how to solve this given the available physical data and statistics. First, the value in E should be all non-negative since it represents the intensity. Second, because the light source point only occupies a small amount of the sub-pixel region even in a high N_{ppp} image, the solution is expected to be sparse. Based on the above reasons the inversion algorithm chose in the paper is a popular non-negative least square solver. The optimization problem is:

$$\text{minimize } \frac{1}{2} \|WE - I\|_2, W \in R^{N \times M}, I \in R^N \quad (12)$$

where N is the number of pixels of I and M is the number of sub-pixel space grids. The NNSL algorithm can be done by MATLAB command *lsqnonneg* which is based on [17]. Although *lsqnonneg* is seen as a greedy algorithm means it is computationally expensive, the accuracy of

the result produced from it is improved significantly from to MCCM and single image support set method.

After getting the vectorized intensity E in a sub-space grid, we can reshape it into the original 2D image size. Then a morphological procedure, dilation, using a square structural element of width L_{sq} is used to accomplish finding local maximum and it is illustrated in Figure 12.

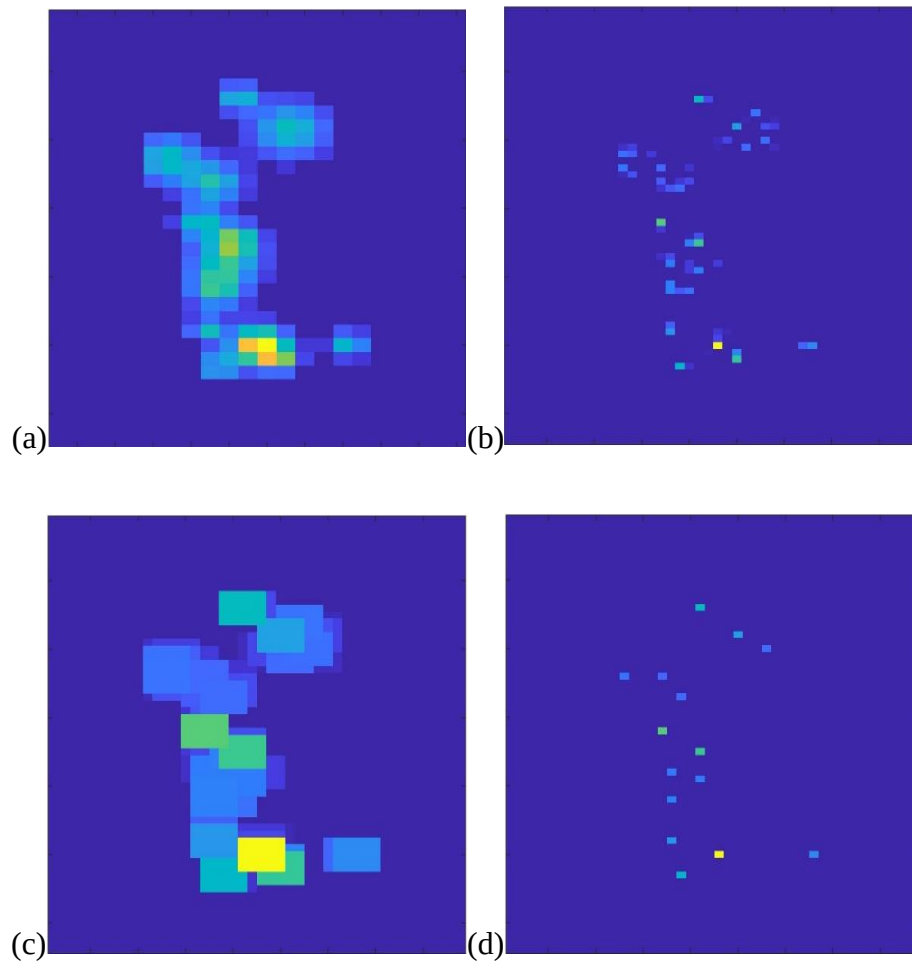


Figure 12. The EDT visual illustration (a) The original image (b) The image from NNLS process (c) Dilation image using square structure element, complete by *imdilate* and *strel* in MATLAB (d) The local maximum found from each dilated area from(c))

Small L_{sq} may lead to more False Positive being unfiltered. In opposite, large L_{sq} can filter out True Positive when there are close particles. The size of L_{sq} is related to size of Δ_E . The larger the grid size, the smaller L_{sq} must be in order not to filter out the close particles. The size effect of L_{sq} is investigated in [16] and in this thesis the use of L_{sq} is as following:

Δ_E	L_{sq}
$1/2$	3
$1/5$	5
$1/10$	7
$1/20$	11

Table 2

3.4 MCCM-NNLS

The NNLS method shows the superior ability in finding particle location in a constant radius image. Nevertheless, if we want to extend the idea and good result from NNLS to size varying image, some serious problems will occur. One problem is that σ_{PSF} is user-defined for NNLS, but for a random particle image, it is difficult to guess size input. Another serious problem is that the dispersion of detection when σ_{PSF} is not determined correctly, which is discussed in Section 4.3.1. To deal with the problems above, we propose a new mixed method MCCM-NNLS. MCCM-NNLS uses MCCM prior to NNLS to get reasonable σ_{PSF} for the inversion process. The process flow is shown in Figure 13, and the details are illustrated in Section 4.3.

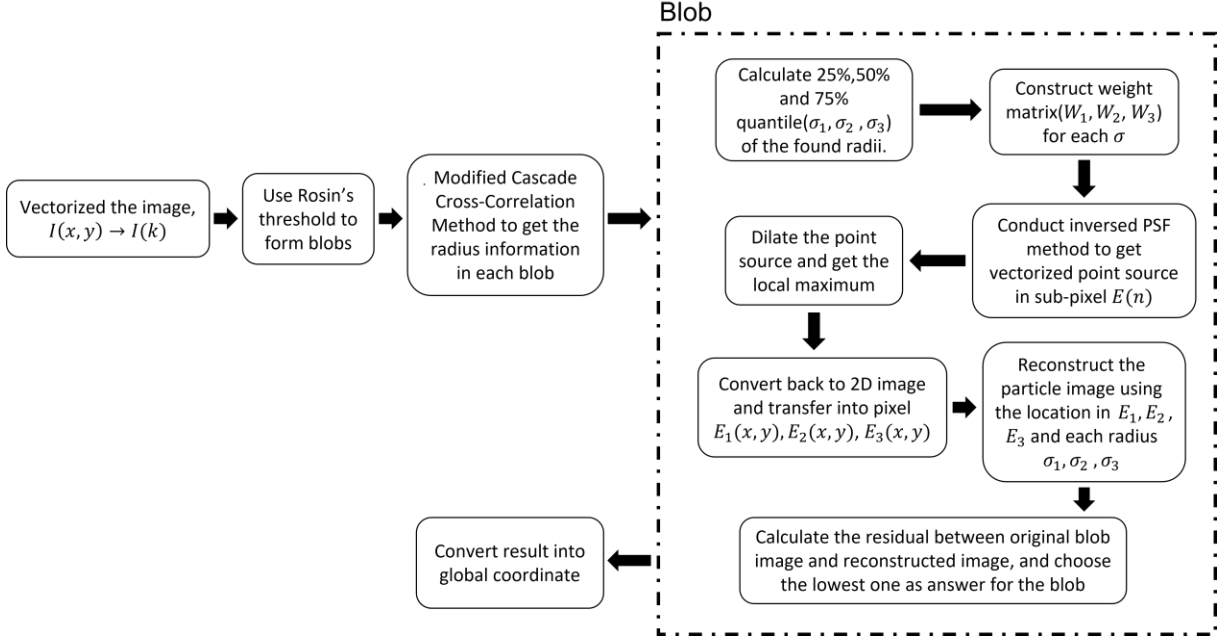


Figure 13. Process flow of MCCM-NNLS method on various radius particle image

SECTION 4. NUMERICAL TESTS RESULTS

The performance of the methods above within representative experimental conditions will be presented in this section.

4.1 Two-particle tests

This test can be used to see how successfully the algorithm separates overlapping particles, which is important for high N_{ppp} images with a lot of overlapping. Due to the discretization nature in pixel space shown in Figure 14, particles at different positions will lead to the histogram change even when they are same particles at same overlap ratio. For each overlap ratio test, 100 images with random locations within 1 pixel will be generated.

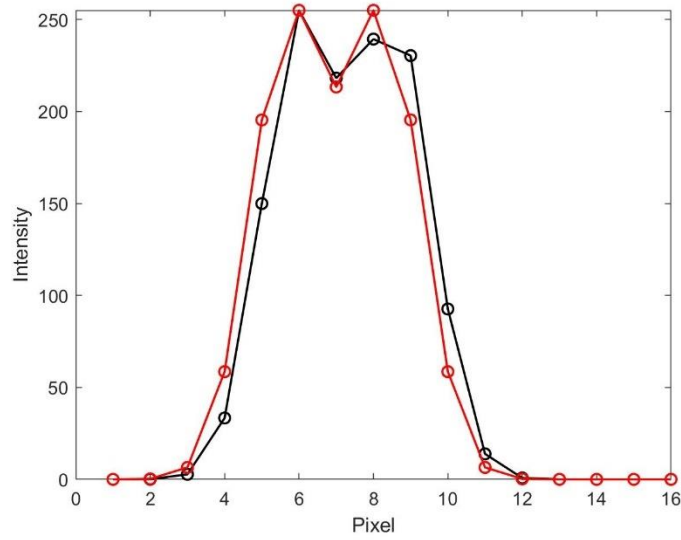


Figure 14. Discretized intensity profile of 2 identical particles($r = 0.6$) with *overlap ratio* = 40% at different sub-pixel center location

The goal of this test is to demonstrate how likely it is for an algorithm to separate particles. We only keep track of the number of particles detected, and yield and reliability aren't considered. It's because if we get the particles close enough together, they'll appear to merge into a single larger particle, and the algorithm will only identify one particle, which isn't a True Positive. Nonetheless, if the particles are brought even closer, the detection will become a True Positive to one of the particles and let yield and reliability rise even it did not separate them. 100% particle found percentage means that the algorithm successfully detects 2 particles in all 100 images, and 50% means only 1 particle detected. The test result of the methods mentioned above is shown in Figure 15.

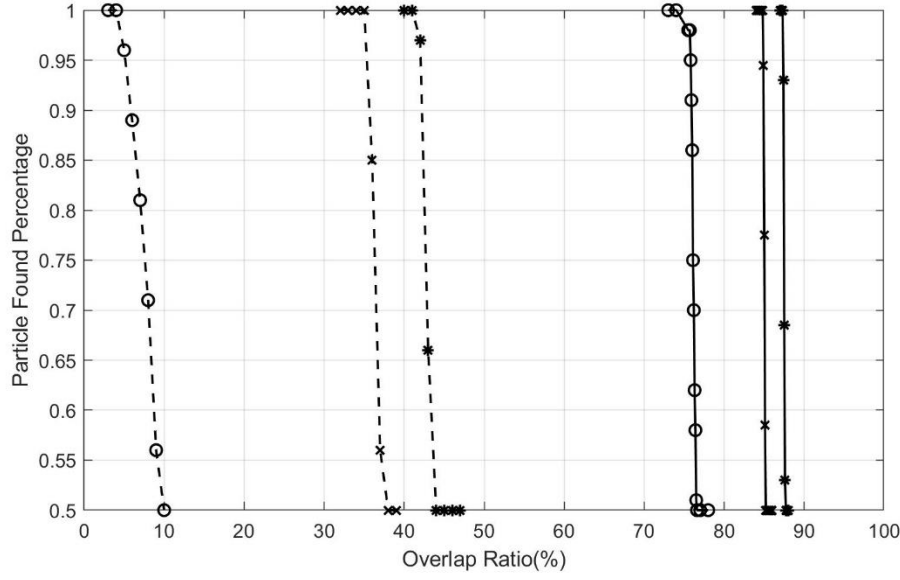


Figure 15. Comparison between NNLS and MCCM in 2 particle overlapping test at $r = 0.6, 1, 1.2$

From the above test result, the limit of NNLS to separate particles is way higher than MCCM, this means that the NNLS has more potential of correctly distinguish particles' location in high seeding density images.

4.2 Constant radius particle tests

In this section, we investigated the constant radius particle images under different density conditions. The test particle radius is set to constant 0.6 pixels. For MCCM, we start with $r = 2$ to $r = 1$ in the cascade process to estimate the particle number N .

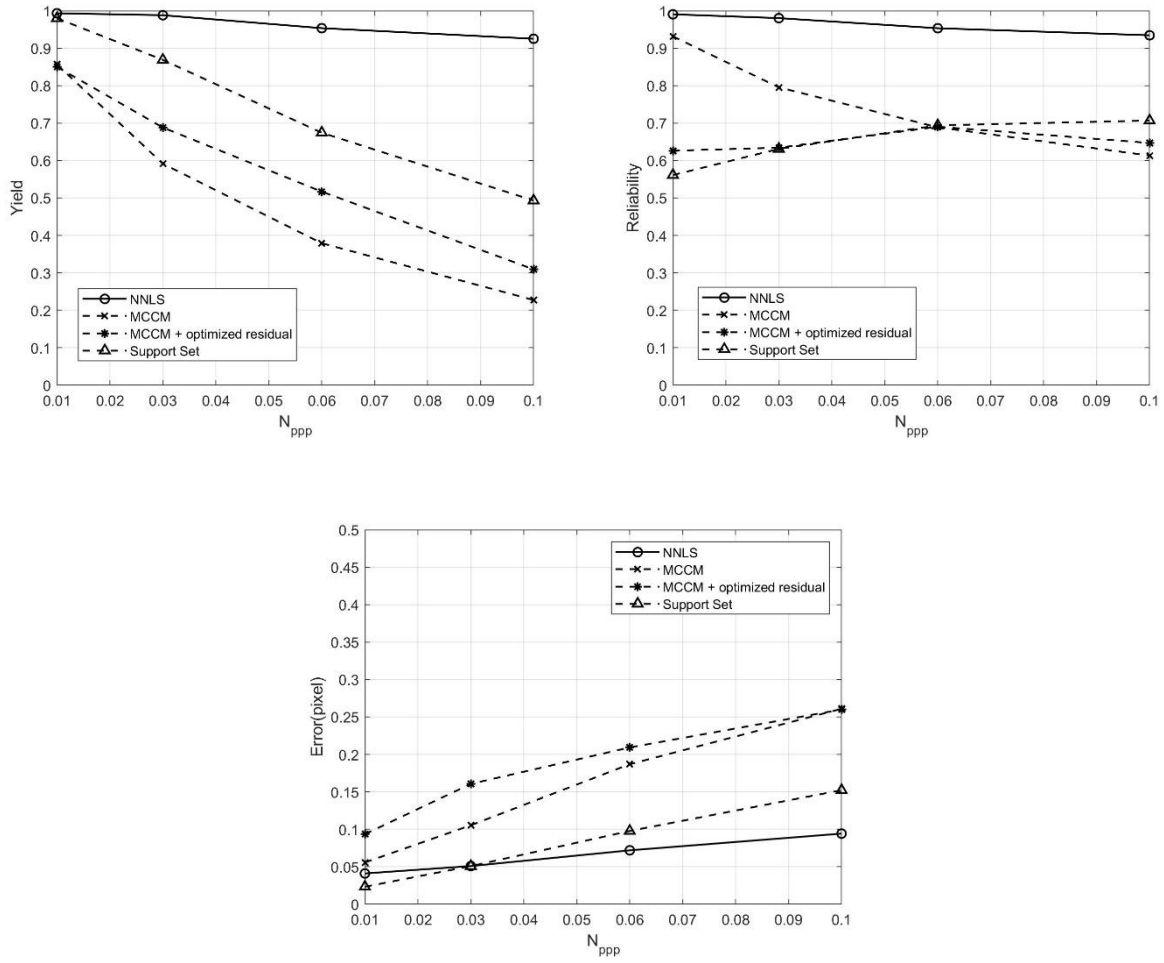


Figure 16. The yield and reliability performance in different particle density conditions between NNLS, MCCM, MCCM + optimized residual and Support Set method at $r = 0.6$

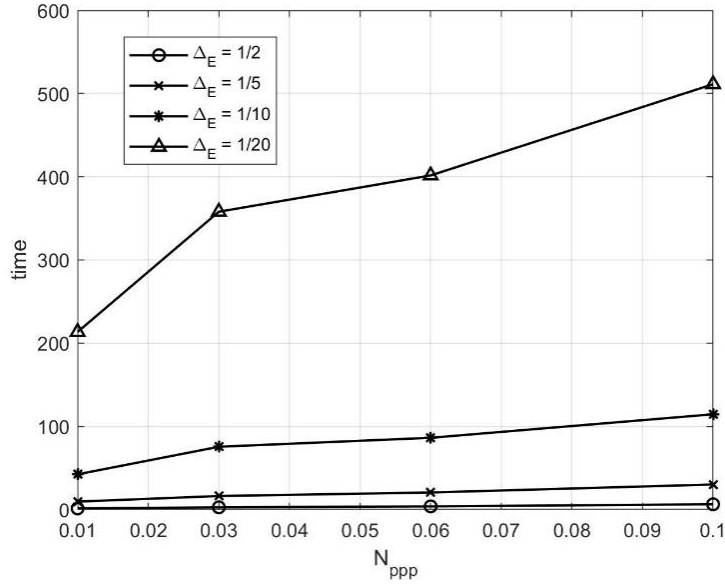


Figure 17. The NNLS method computational time divided by MCCM computational time of different Δ_E under different N_{ppp} . The hardware spec is Dell XPS15 Intel® CPU i7-7700 @ 2.80GHz with 16GB RAM

As we observed in Figure 16 the ability of the NNLS method to accurately detect particle location is superior to other methods under any N_{ppp} condition. However, Figure 17 shows the required computational resource comparison that NNLS takes the longest time to get the result. Before implementing the MCCM-NNLS method that targets to deal with various size particle images, it is tested on the constant radius image. The result is shown in Figure 18.

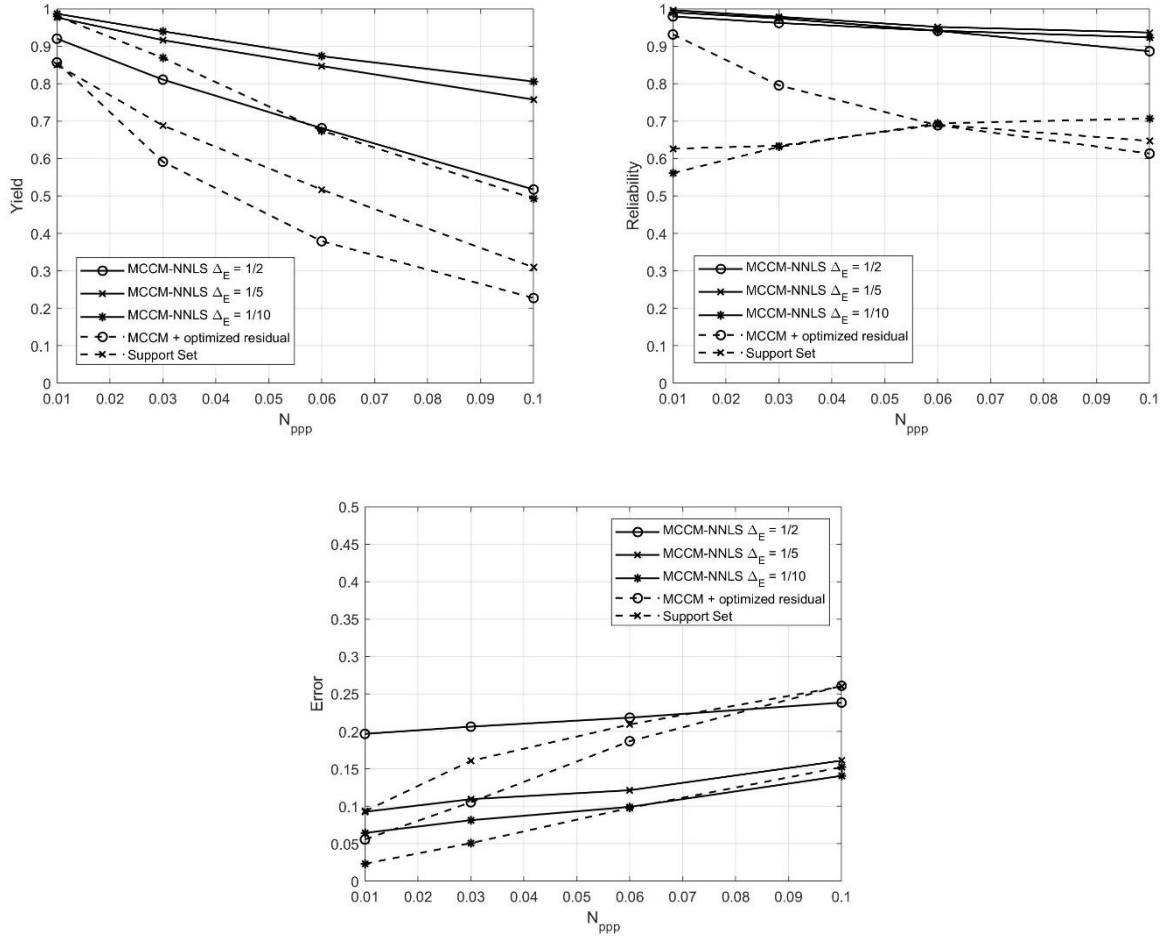


Figure 18. The performance of the proposed technique MCCM-NNLS on $r = 0.6$ image

From the above figures, we can preliminarily observe that the MCCM-NNLS method works on constant radius images. The reason that Δ_E are chosen down to only $\frac{1}{10}$ is due to the limit of computational resources. In the MCCM-NNLS approach, the weight matrix building, and nonnegative least-square process must be done 3 times in each blob. Despite this, the outcome is still superior to the other approaches shown in Figure 18. As a result, we now have a foundation on which to build varied radius images.

4.3 Various radius particle tests

4.3.1 Impact of inaccurate σ_{PSF}

The prior methods were all examined on a synthetic particle image with a constant intensity and radius, which is set to be 0.6 pixels. As said earlier in this thesis, $r = 2$ to $r = 1$ in cascade cross-correlation step is used in MCCM, $r = 0.6$ is used in support set method, and the NNLS method uses the exact input $\sigma_{PSF} = 0.6$ as the input for PSF to get the primary result of the algorithms. The outcome from the Figure 16 is undoubtedly the best among the three methods by a huge margin especially at high-density conditions and might not have much room for improvement in terms of yield, reliability, and accuracy. However, in the true 2D PTV image, the size particle size may vary due to some reason such as the out-of-focus effect while taking pictures. Figure 19 shows how the choice of σ_{PSF} would affect the precision of particle detecting where the real particle radius is set to 0.6 pixels. If σ_{PSF} underestimates the real particle size, such as $\sigma_{PSF} = 0.4$, the inverse step will find many False Positive that significantly lowers the reliability and Figure 20 is the qualitative view of the situation. This is due to the inversion process tries to rebuild a large-area particle picture using small-area particles. Because the imaging model uses a summation of particle pictures, the program artificially generates particles whose signal intensities add to the signal intensity distribution of the big particle (in the least square sense) as the radius is misjudged. This observation was also discussed in [16]. On the other hand, the overestimation of σ_{PSF} in a clustered region may lead to the rise of False Negative which will have a negative impact on the performance of the algorithm especially in terms of yield.

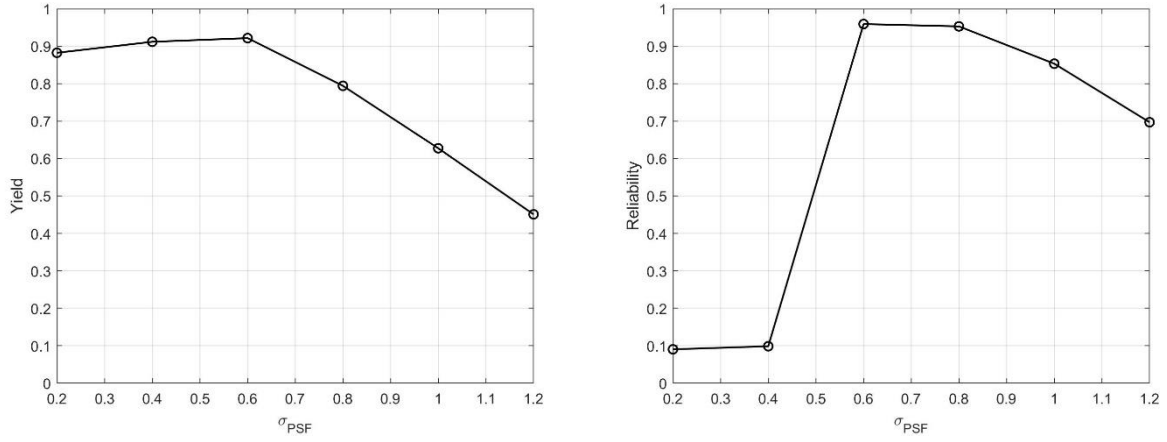


Figure 19. Yield and reliability result when applying different σ_{PSF} on 30 images that $r = 0.6$,

$$N_{ppp} = 0.1$$

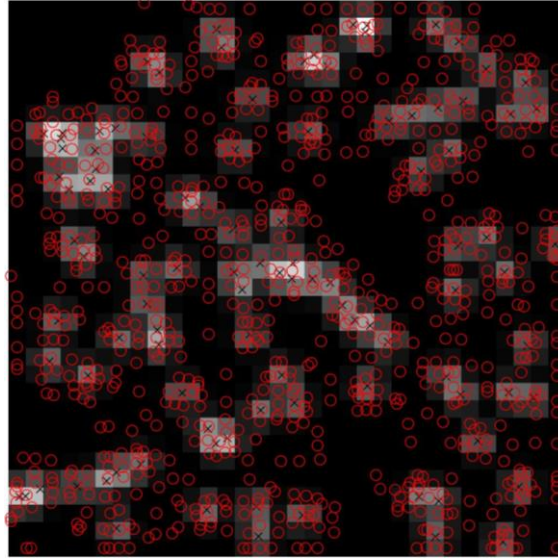


Figure 20. The qualitative view of the impact of underestimating the real size of particle while using NNLS($r = 0.6, \sigma_{PSF} = 0.4$). The red circles are the found particle location and most of them are False Positive.

Therefore, the estimation of σ_{PSF} is crucial for the NNLS method in various radius images. In this section, the random distribution of particle size from and (1) 1 pixel to 1.2 pixels (2) 0.6

pixel to 1.4 pixels is generated. 30 photos are generated at random for each case. Since we know the importance of estimation of σ_{PSF} , next step is to find the optimized value of it in varying size images. The idea here is to utilize MCCM technique to obtain a reasonable approximation. The non-linear least-squares fitting solver can give us the information about x,y location, intensity, and the radius of particles and we take advantage of the fitted radius. We calculate the mean radius of all particles found in MCCM to be the weight matrix input σ_{PSF} for the NNLS method. This process helps us obtain a reasonable guess for the inversion step and the yield rise about 25% and 10% of reliability when $N_{ppp} = 0.1$ which is shown in Figure 21.

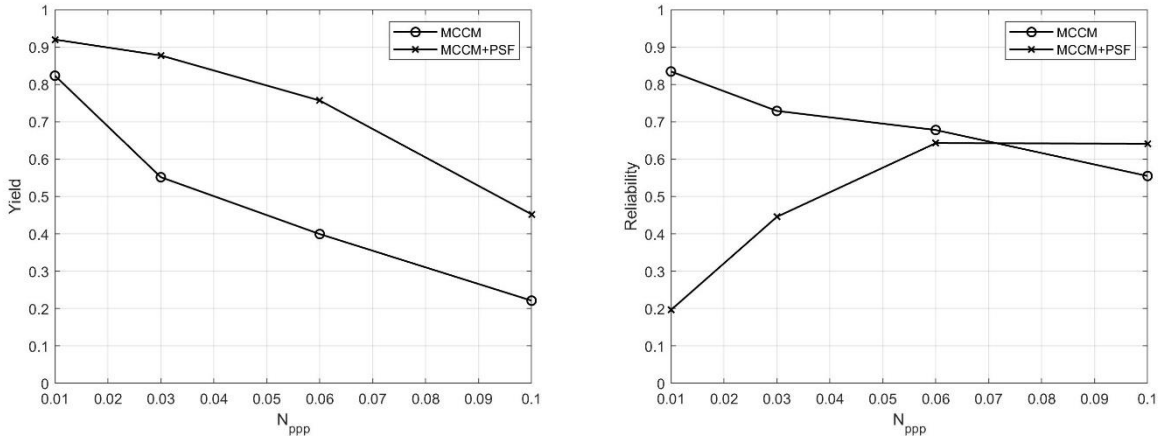


Figure 21. Result of taking the median of found radius from MCCM to be σ_{PSF} in $r = 0.6$ – 1.4 image

However, when the image is not dense, the reliability drops dramatically. Figure 22 illustrates why this happens. If there are two different size particles, using median found radius as σ_{PSF} will result in False Positive dispersion because one of the particle sizes is underestimated and the same problem will occur if we pick the minimum found radius. On the other hand, if we select the maximum found radius as σ_{PSF} , the NNLS algorithm may not detect the particles that are too close. To address this problem, instead of median radius we found in MCCM, we also try to take

25% and 75% quantile of all found particle radius as σ_{PSF} input of PSF. The reason for taking these 3 values is to try to find optimal σ_{PSF} in every situation, it is $N_{ppp} = 0.01 - 0.1$ here. In Figure 23 we can notice that when the density is low MCCM can almost perfectly find the correct radius. Yet when N_{ppp} goes up, MCCM tends to find larger particle than actual size because of the cluster of particles. Combining the two arguments above, 25%, 50%, and 75% quantile of size detected by MCCM in each blob are chosen. After the inversion part, we reconstruct the image with each σ_{PSF} as radius then calculate the residual between it and the original blob image. The σ_{PSF} with the lowest residual is chosen and the particle location found by it consider as the result. Although this step may cause the more computational time to complete, the improvement of the outcome can be easily seen in Section 4.

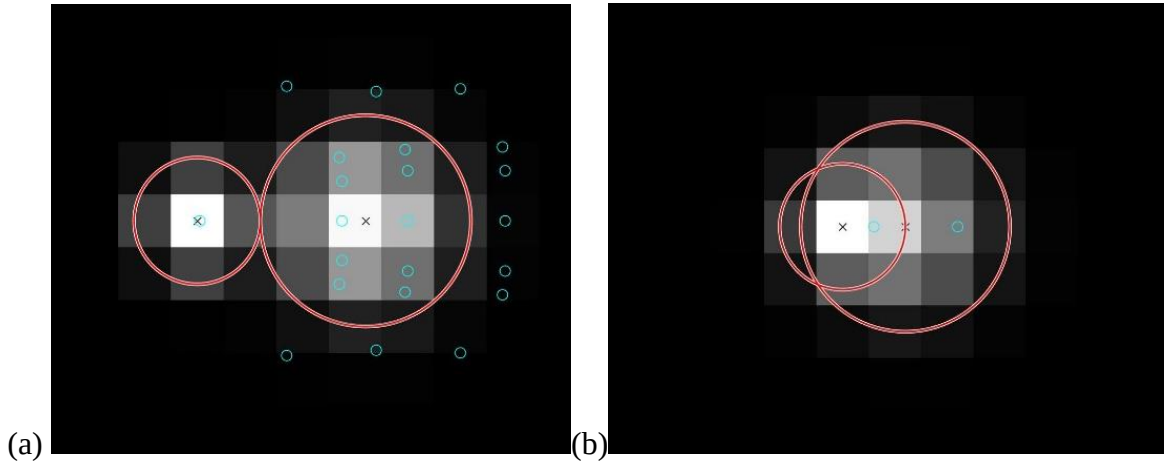


Figure 22. Qualitative view of $r = 0.6$ and $r = 0.8$ particles at (a) *overlap ratio* = 20% applying $\sigma_{PSF} = 0.6$ (b) *overlap ratio* = 70% applying $\sigma_{PSF} = 1$, the blue circles are the found particle location by NNLS

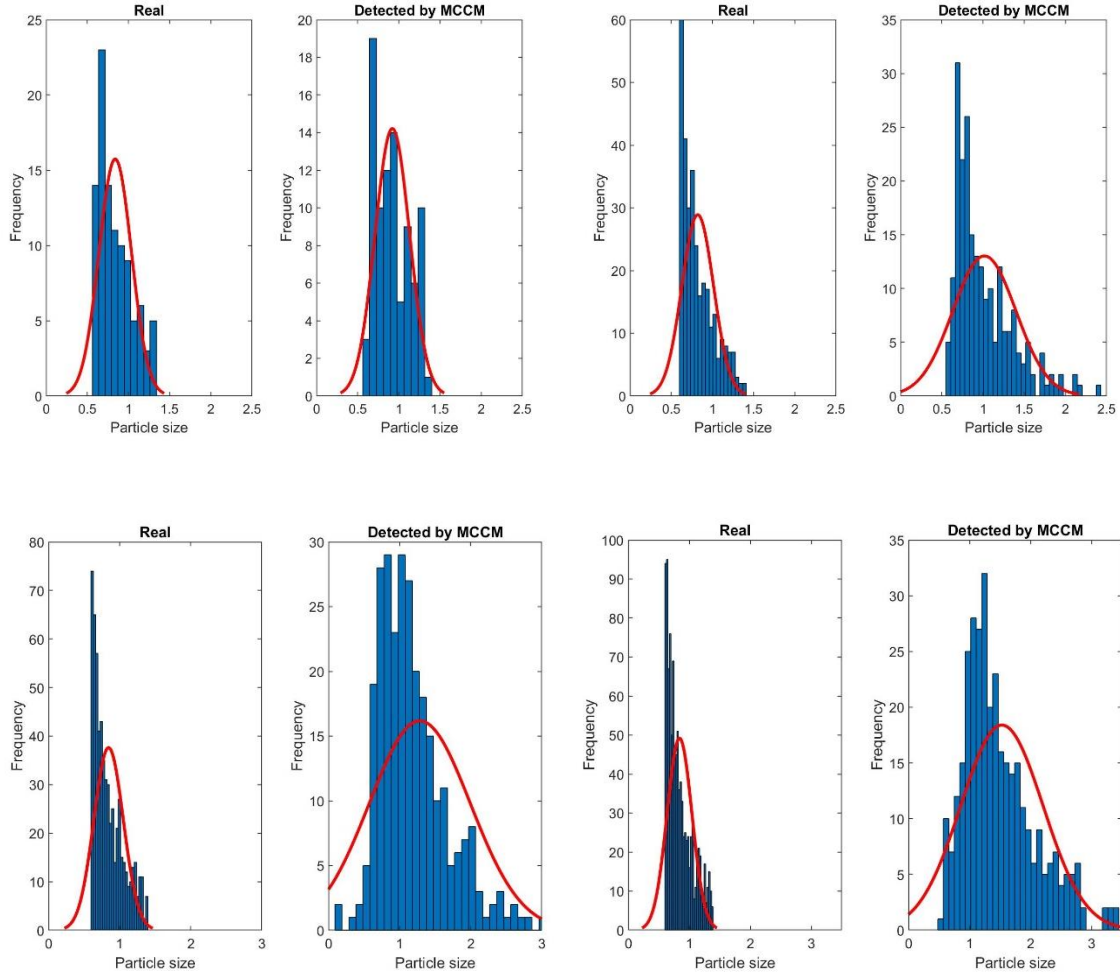


Figure 23. The normal distribution fit of real particle size vs particle size detected by MCCM at

$$r = 0.6 - 1.4$$

4.3.2 The effect of Δ_E

The grid size Δ_E can determine the accuracy of found peaks in constant size particle images. The finer the grid is, the better accuracy it can reach in compromise with computational time which is discussed in [16]. In the varying size image, σ_{PSF} is not perfect and may cause the disperse of False Positive like Figure 22 even we have tried to optimize our choice in previous section. Thus, we investigate how different Δ_E affect the dispersion of False Positive in Figure

24. From the result, we can observe that the dispersion is more severe when Δ_E is smaller which will have a huge impact on reliability.

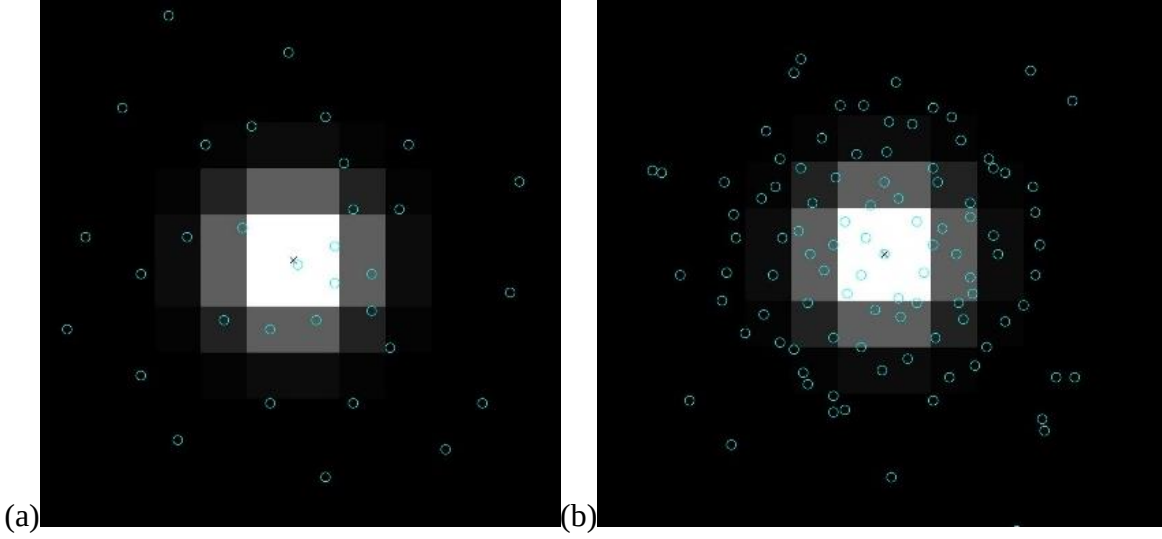


Figure 24. Qualitative view of dispersion of False Positive. $r = 1$, $\sigma_{PSF} = 0.6$, (a) $\Delta_E = 1/5$ (b)

$$\Delta_E = 1/20$$

Thus, we tried $\Delta_E = \frac{1}{2}, \frac{1}{5}, \frac{1}{10}$ to do the MCCM-NNLS method. The improvement made by switching MCCM to MCCM-NNLS can be observed in Figure 25, Figure 26, and Figure 27. For $r = 1 - 1.2$, the yields and reliability at all Δ_E are almost all better than MCCM. The yield goes up approximately 30% for $\Delta_E = \frac{1}{20}$ at $N_{ppp} = 0.1$ and even increases more than 50% at $N_{ppp} = 0.06$. Reliability also raises 10% to 20% if we choose the finer grid. On the other hand, despite when N_{ppp} is low, the performance of proposed method in $r = 0.6 - 1.4$ is about 5% to 7% lower but is still considered high, and in a higher density situation, the MCCM-NNLS method has up to 35% better yield and 25% better reliability performance.

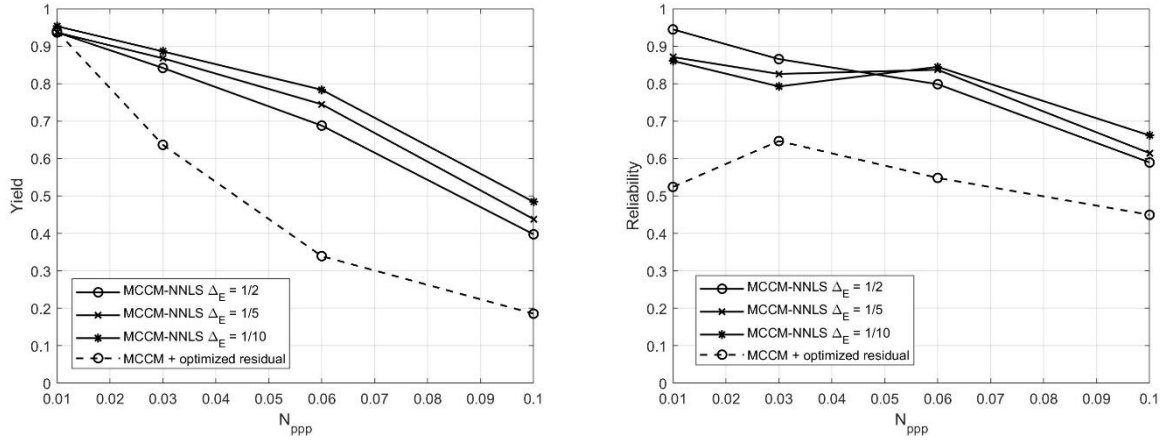


Figure 25. Performance comparison between MCCM + optimized residual and MCCM-NNLS at

$r = 1 - 1.2$ image

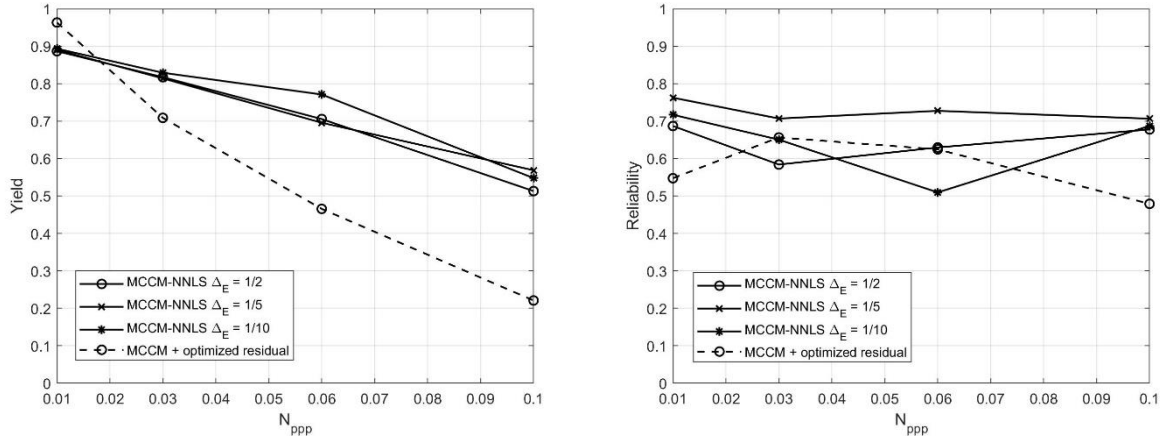


Figure 26. Performance comparison between MCCM + optimized residual and MCCM-NNLS at

$r = 0.6 - 1.4$ image

For the error part in Figure 27, the proposed method is not as good as MCCM at low seeding density, which is foreseeable. The NNLS can only fit a single size particle into an image while MCCM takes the size as a variable that can find the most suitable size for found particles thus lower the location error. Another reason is that the found positions in the NNLS method are constraint to discrete sub-pixel, in comparison to the continuous non-linear least-squares fitting

the error drops only when the grid size goes down. Nevertheless, the error of MCCM increases more quickly than the MCCM-NNLS method along with the rising density, and at the $N_{ppp} = 0.1$ the error is almost the same for both methods. Therefore, we can say that MCCM-NNLS outperforms the MCCM in terms of yield, reliability, and error in various size particle conditions ($r = 1 - 1.2$ and $r = 0.6 - 1.4$), especially at a high seeding density N_{ppp} .

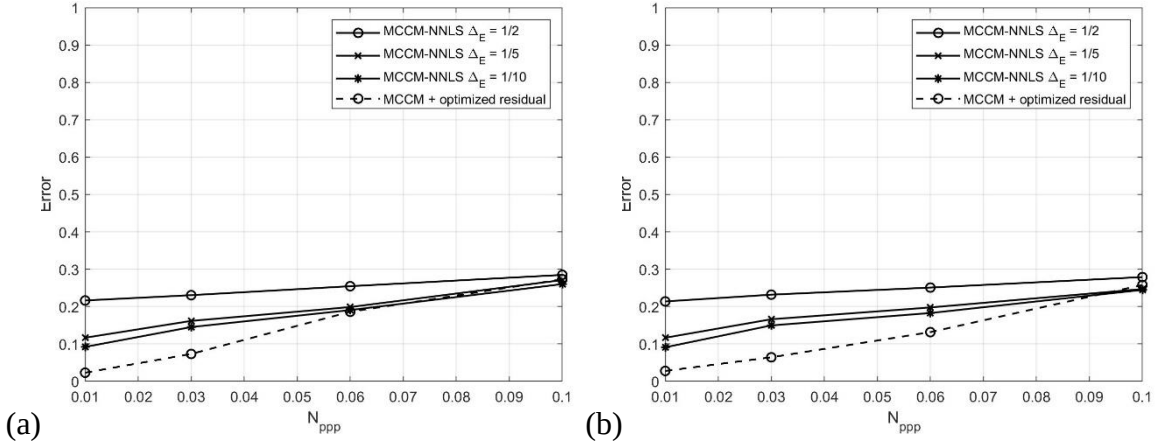


Figure 27. Error comparison between MCCM-NNLS with different and MCCM + optimized residual at (a) $r = 1 - 1.2$, (b) $r = 0.6 - 1.4$

SECTION 5. CONCLUSION AND FUTURE WORK

The proposed method that mixed MCCM and NNLS has the superior ability in terms of yield and reliability to distinguish particle peak location in various size images. We had examined in the synthetic image that $r = 1 - 1.2$ and $0.6 - 1.4$ with constant intensity. The tests that involve more diverse sizes and intensities can be investigated. For noise-free and constant radius particle images, if the standard deviation of NNLS is defined precisely enough, yield and reliability can even go up to 95%. On the other hand, if we consider a more realistic situation that the radius varies, the proposed way is to mix MCCM and NNLS methods. MCCM can be used to

determine suitable σ_{PSF} guess for NNLS and NNLS can provide a more accurate result than using MCCM along. The examined result in Section 4. shows that the yield and reliability can be way higher especially at high seeding density. However, this kind of method is very time-consuming. As mentioned prior in Section 3.3, *lsqnonneg* is a greedy algorithm. A good result requires fine grids to accomplish in high N_{ppp} situation, which causes the weight matrix to become huge. For example, the size of weight matrix W the 32×32 pixels image with $\Delta_E = \frac{1}{20}$ is 1024×409600 . The typical 2D PTV image size is thousands by thousands of pixels, the weight matrix size for this would be millions by tens of thousands, which is a huge inversion job for *lsqnonneg*. There have already been several ways to ease the inversion process and therefore increase the efficiency, such as FNNSL[18], but they are not yet implemented in MATLAB, that is the reason we stick with classic NNLS in this thesis. Another future that can be done is to take noise into consideration. The results from this thesis are all conducted in synthetic images that do not include noise. The original paper [5] and [16] did include the noise in their images, giving the high hopes that the proposed method can also be applied in a more realistic condition which needs to do further investigation.

BIBLIOGRAPHY

- [1] K. Ohmi and H. Y. Li, “Particle-tracking velocimetry with new algorithms,” *Meas. Sci. Technol.*, vol. 11, no. 6, pp. 603–616, 2000, doi: 10.1088/0957-0233/11/6/303.
- [2] M. R. Brady, S. G. Raben, and P. P. Vlachos, “Methods for Digital Particle Image Sizing (DPIS): Comparisons and improvements,” *Flow Meas. Instrum.*, vol. 20, no. 6, pp. 207–219, 2009, doi: 10.1016/j.flowmeasinst.2009.08.001.

- [3] ا. سینا, “No Titleطب”, p. 283, 1386.
- [4] N. C. Angarita-Jaimes, M. G. Roca M., C. E. Towers, N. D. Read, and D. P. Towers, “Algorithms for the automated analysis of cellular dynamics within living fungal colonies,” *Cytom. Part A*, vol. 75, no. 9, pp. 768–780, 2009, doi: 10.1002/cyto.a.20750.
- [5] Y. C. Lei *et al.*, “A vision-based hybrid particle tracking velocimetry (PTV) technique using a modified cascade correlation peak-finding method,” *Exp. Fluids*, vol. 53, no. 5, pp. 1251–1268, 2012, doi: 10.1007/s00348-012-1357-6.
- [6] B. Wieneke, “Iterative reconstruction of volumetric particle distribution,” *Meas. Sci. Technol.*, vol. 24, no. 2, 2013, doi: 10.1088/0957-0233/24/2/024008.
- [7] T. Jahn, D. Schanz, and A. Schröder, “Advanced iterative particle reconstruction for Lagrangian particle tracking,” *Exp. Fluids*, vol. 62, no. 8, pp. 1–24, 2021, doi: 10.1007/s00348-021-03276-7.
- [8] P. Ruhnau, C. Guetter, T. Putze, and C. Schnörr, “A variational approach for particle tracking velocimetry,” *Meas. Sci. Technol.*, vol. 16, no. 7, pp. 1449–1458, 2005, doi: 10.1088/0957-0233/16/7/007.
- [9] N. Otsu *et al.*, “Otsu_1979_otsu_method,” *IEEE Trans. Syst. Man. Cybern.*, vol. C, no. 1, pp. 62–66, 1979.
- [10] P. L. Rosin, “Unimodal thresholding,” *Pattern Recognit.*, vol. 34, no. 11, pp. 2083–2096, 2001, doi: 10.1016/S0031-3203(00)00136-9.
- [11] N. F. Ponchaut, “Part I : 3DPTV : Advances and Error Analysis Part II : Extension of Guderley ’ s Solution for Converging,” *Analysis*, vol. 2005, 2005.

- [12] A. Boomsma and D. Troolin, “Time-Resolved Particle Image Identification and Reconstruction For Volumetric 4D-PTV,” *19th Int. Symp. Appl. Laser Imaging Tech. to Fluid Mech.*, p. paper 246, 2018.
- [13] N. D. Cardwell, P. P. Vlachos, and K. A. Thole, “A multi-parametric particle-pairing algorithm for particle tracking in single and multiphase flows,” *Meas. Sci. Technol.*, vol. 22, no. 10, 2011, doi: 10.1088/0957-0233/22/10/105406.
- [14] G. E. Elsinga, F. Scarano, B. Wieneke, and B. W. Van Oudheusden, “Tomographic particle image velocimetry,” *Exp. Fluids*, vol. 41, no. 6, pp. 933–947, 2006, doi: 10.1007/s00348-006-0212-z.
- [15] F. Champagnat, P. Cornic, A. Cheminet, B. Leclaire, G. Le Besnerais, and A. Plyer, “Tomographic PIV: Particles versus blobs,” *Meas. Sci. Technol.*, vol. 25, no. 8, 2014, doi: 10.1088/0957-0233/25/8/084002.
- [16] A. Cheminet, J. F. Krawczynski, and P. Druault, “Particle image reconstruction for particle detection in particle tracking velocimetry,” *Meas. Sci. Technol.*, vol. 29, no. 12, 2018, doi: 10.1088/1361-6501/aae9c2.
- [17] “Solving least squares problems C. Lawson, R. Hanson 1974 vol 15 (Upper Saddle River, NJ: Prentice-Hall) ch 23 p 161,” vol. 15, p. 1974, 1974.
- [18] R. Bro and S. de Jong, “A Fast Non-negative Constrained Linear Least Squares Algorithm,” *J. Chemom.*, vol. 11, no. February, pp. 393–401, 1997.