Motion blur treatment utilizing deep learning for time-resolved particle image velocimetry

Jeong Suk Oh¹ · Hoonsang Lee¹ · Wontae Hwang¹

Received: 21 May 2021 / Revised: 10 September 2021 / Accepted: 10 October 2021 / Published online: 21 October 2021
© The Author(s) 2021

Abstract
A new method is hereby presented to reduce motion blur induced error of time-resolved particle image velocimetry. The Monte-Carlo method (MCM) was applied to synthetic images to quantify the error due to blurred particle images. As the size of the streaks grew, it caused large errors in estimating displacements and increased the frequency of outliers beyond 20% for some cases. The mean displacement error was also about 0.2 – 0.55 px, which is larger than the nominally accepted PIV uncertainty of 0.1 px. A novel deblur filter (i.e., the generator) using a generative adversarial network (GAN) was developed, using 1 million synthetic images. The generator was verified using unlearned data from the MCM. The frequency of outliers, which was originally higher than 20% for the worst case, decreased to about 6%, and the displacement error was reduced to less than 0.3 px. The generator was applied to actual experimental images of a synthetic jet that had image blur and resulted in a substantial reduction of outliers. We also checked the performance of the generator in a uniform channel flow, and found that the deblurred images resulted in less PIV velocity error, and was closer to the results from the sharp images than those from the blurry images.

Graphic abstract

---

* Wontae Hwang
wthwang@snu.ac.kr

Extended author information available on the last page of the article
1 Introduction

Time-resolved particle image velocimetry (TR-PIV) utilizes high-speed photography to evaluate continuous fluid motion in a non-intrusive manner. Unsteady flow phenomena such as turbulence can be measured with this technique. Research has been ongoing to improve the capability of TR-PIV (She et al. 2021; Beresh et al. 2021). The measurement procedure can be broken up into image pre-processing, PIV analysis, and post-processing. The main focus of this study is the pre-processing step. In many cases, image quality may not be optimal due to various limitations during image acquisition (Rossi et al. 2012). To mitigate such issues, image pre-processing techniques can be employed to improve signal-to-noise ratio (SNR), as described by Mendez et al. (2017) and Zhu et al. (2013).

Various reasons can affect image quality, which can result in PIV error. Bugg and Rezkallah (1998) showed that image background noise causes displacement error. Scharnowski and Kähler (2016b) noted that this is due to loss of correlation. To reduce image noise, Vétel et al. (2011) utilized various denoising filters, and confirmed that the Wiener filter effectively eradicates noise and reduces error. Sciacchitano and Scarano (2014) were also able to eliminate reflective light within the images using a high-pass filter. In addition to image pre-processing, many studies have applied post-processing after vector calculation (Nogueira et al. 1997; Seol and Socolofsky 2008; Stanislas et al. 2003), but these techniques are not the focus of this study and therefore will not be discussed here.

Motion blur, which brings about particle streaks in the images, arises when the flow is relatively fast compared to the image exposure time and results in additional error (Grayver and Noir 2020). The region of interest in TR-PIV is often fairly small when resolving small-scale turbulence and in this case moving particles can easily create streaks. This tendency is more pronounced with the use of a continuous wave (CW) laser, which is often preferred over high repetition rate pulsed lasers, due to the significantly lower cost (Neeraj and Lal 2013). The streaks can be reduced by decreasing the exposure time of the camera, but then the light intensity from the particles becomes weak.

Previously, there have been various discussions regarding motion blur issues. Wang et al. (2018) observed the wake behind a circular cylinder utilizing TR-PIV with a CW laser. They compared the results with those obtained from conventional PIV. They indicated that erroneous turbulence statistics resulted from particle streaks. Elzawway (2012) showed that an increase in streak size due to increase in exposure time when using a CW laser results in an increase in the displacement error.

Although motion blur greatly affects PIV results, there have not been many quantitative analyses discussing its effect. Elsinga et al. (2005) conducted a study analyzing how the cross-correlation map is affected by blur due to the refractive index difference near a shock wave. In this study, the cross-correlation map was asymmetric because of the blur effect. They noted that this could cause errors in the PIV analysis. Although the root cause of blur generated in this study was not due to motion, the blur pattern is similar. It is still not exactly known how much error image blur can produce, and efforts to fully resolve this issue do not yet exist.

There has not been much research that aims at resolving the PIV image streak issue within the PIV community, thus we investigated studies that have tried to eradicate blur in other disciplines. Image blur itself is a classical problem, and many studies have been conducted in an effort to solve it. Traditionally, image deblur was based on approaches such as iterative methods (Biemond et al. 1990). With the advent of deep learning, many current deblurring methods now employ this technology. Svoboda et al. (2016) compared and analyzed a traditional deblurring method ( Blind L0) with a convolutional neural network (CNN) L-15. A comparison of blurred license plates induced by car movement revealed overwhelming performance of CNN L-15. Kupyn et al. (2018) attempted to compare results of the existing CNN method with their method that uses Generative Adversarial Networks (GAN), which was initially proposed by Goodfellow et al. (2014). The study showed that GAN removes blur from images more effectively than conventional CNN. Therefore, this suggests that this type of deblurring could possibly be implemented for PIV.

This study focuses on quantifying the blur effect and developing a deblur filter which can reduce blur induced PIV errors. It contains three main sections, as depicted in Fig. 1. Section 2 identifies the extent of PIV error caused by image blur via the Monte-Carlo method, using synthetic images. Section 3 describes the deblur filter development using the GAN method, and its performance. Section 4 demonstrates application of the deblur filter in an actual PIV experiment of a synthetic jet.

2 Blur effect quantification

2.1 Synthetic image

The blur effect is first quantified via synthetic PIV images, which allows easy control of various parameters such as particle diameter, number density, and displacement. Synthetic images have already been used in various studies, and it has been verified that these images sufficiently represent real cases (Scharnowski and Kähler 2016a, b; Sciacchitano 2019; Sciacchitano et al. 2012; Lee and Hwang 2019).
The image intensity of particles follows the Airy function, which can be approximated by a Gaussian profile (Adrian and Yao 1985). Lecordier and Westerweel (2004) produced synthetic images using 2D Gaussian profiles for the particles. To convert a continuous distribution into discrete pixels of the image, light flux within the area corresponding to a single pixel of the continuous 2D Gaussian distribution was integrated. The pixel intensity value \( I \) at location \((x, y)\) due to particle light scattering is expressed as follows:

\[
I(x, y) \propto d_p^2 \int_{-\frac{x_f}{2}}^{\frac{x_f}{2}} e^{-\frac{1}{2}(x-x_p)^2} dx \int_{-\frac{y_f}{2}}^{\frac{y_f}{2}} e^{-\frac{1}{2}(y-y_p)^2} dy
\]  

where \( d_p, f_x, f_y, x_p, \) and \( y_p \) are the diameter, pixel width in \( x \) and \( y \) direction, and center positions of the particle, respectively. After the synthetic image is generated, a blur filter is applied to produce a blurry image with particle streaks. Schuon and Diepold (2009) have shown through experiments that synthetic motion blur is representative of actual motion blur. The motion blur filter often utilizes a point spread function (PSF), and the 2D square synthetic PSF can be expressed as Eq. (2)

\[
h_{2D} = \frac{1}{K} \begin{bmatrix} h_{1,1} & \cdots & h_{1,k_2} \\ \vdots & \ddots & \vdots \\ h_{k_2,1} & \cdots & h_{k_2,k_2} \end{bmatrix}
\]  

where \( K \) is a normalizing constant which is the sum of all components inside the \( h_{2D} \) matrix, and \( k_2 \) is the filter size which indicates the magnitude of motion. The \( h_{2D} \) matrix was obtained using a MATLAB function to simulate motion blur, by specifying the angle and size of the blur. Each element of the matrix was determined by calculating the nearest distance between its location and the virtual line with the specified angle and length. As the blur filter size increases, the size of the particle streak also increases. The blur filter was implemented on the original image via the convolution operator (represented by \( \ast \)) to obtain the blurred image, as shown in the following equation:

\[
b = h_{2D} \ast o + n
\]  

where \( b, o, \) and \( n \) indicate blurred image, original image, and additive noise, respectively.

We applied several blur filters to a single particle, and the results from three different blur filter sizes \((k_2 = 3, 5, 7 \text{ px})\) are compared in Fig. 2 a. It should be noted that these black and white particle images are inverted for enhanced visibility. When the filter was applied, circular particles with a Gaussian intensity profile were stretched to an oval shape in the blur direction. Also, the particle intensity was diffused within the stretched area, reducing the intensity at the center. This effect was more prominent as the filter size increased, as shown in Fig. 2b. The width of the normalized intensity profile at \( e^{-2} \) of the peak brightness, denoted by the gray circles in each profile, corresponds to the particle diameter (Scharnowski and Kähler 2016a, b). Depending on the filter size, the particle size more than doubled in some cases. Since the area under each graph is the same due to the convolution operation of the filter, an increase in the particle size leads to a decrease in peak intensity. The flat plateau in the center is due to the blur filter being uniformly applied in a certain direction.

Another important point is that in the procedure of creating synthetic images, relatively small 32 px \( \times \) 32 px images were used. This reduced image size not only

---

**Fig. 1** Schematic diagram of the overall research
increases the amount of learning possible, but also has other advantages as well. We can assume constant velocity within the exposure time for the 32 px × 32 px image, which corresponds to a standard PIV subwindow (Raffel et al. 2018). In other words, we neglect local acceleration or rotation, and only consider linear motion blur. Thus, we can employ the deblur filter developed in this study and assume that the average particle location is at the center of the streak. Additionally, the deblurring filter developed in this study can be considered universal and used for any type of flow.

We did not consider image intensity as a separate variable when designing the blur filter, even though an intensity increase is expected due to longer exposure times. The reason for this was to limit the number of parameters and not overburden the learning process when creating the deblur filter. It should be noted that the pre-processing step outlined below converts images with larger intensity nearly back to the original images, for both blurred and deblurred cases. Therefore, the results in this study should not have a significant impact due to intensity increase from long exposure times.

Next, we created a synthetic image that contains many particles, and blurred it, simulating a real PIV image. Figure 3 shows a comparison between real PIV images and synthetic images. The sharp images look quite similar to each other, and the blurred images are also similar in appearance. This suggests that, as noted in previous studies (Scharnowski and Kähler 2016a, b; Schuon and Diepold 2009; Sciacchitano 2019; Sciacchitano et al. 2012), we can fully represent the real case with synthetic images.

### 2.2 PIV analysis

We used the MATLAB-based open-source software PIVlab for PIV analysis (Thielicke and Stamhuis 2014). The pre-processing method in the tool was slightly modified for this study. The pre-processing consists of contrast limited adaptive histogram equalization (CLAHE), high-pass filter, and Wiener filter. In the process of determining the displacement of particles, cross-correlation based on Fourier transform was used, which is calculated as follows:

\[
C(x, y) = \mathcal{F}^{-1}\left\{\mathcal{F}(I_1(x, y)) \times \mathcal{F}(I_2(x, y))\right\}
\]  (4)
where $F$ and $I$ represent the Fourier transform and image intensity matrix, respectively. Subscripts 1 and 2 indicate the first and second frame. For sub-pixel accuracy, the final displacement was obtained using the 2D Gaussian sub-pixel estimator suggested by Nobach and Honkanen (2005). It uses the peak in $C(x, y)$ and the eight surrounding neighbors as sub-pixel estimators to locate the exact peak.

There is a possibility that irregular particle scattering intensity profiles can lead to errors in the cross-correlation calculation (Elsinga et al. 2005; Elzawawy 2012). We calculated the displacement between two interrogation windows based on the cross-correlation peak value. When analyzed using the original interrogation windows without blur, the cross-correlation map appeared to be Gaussian in shape because distributions of the particles are Gaussian (Dabiri 2006). Cross-correlation maps were also obtained for blurred windows, as illustrated in Figs. 4 and 5. The results are divided into these two figures depending on whether the sub-pixel estimator worked well or not, for cross-correlation maps with (a) multiple smooth peaks and (b) multiple sharp peaks. The correlation maps are normalized with a peak value of 255. The graphs in the left column represent the discrete correlation map, and the ones on the right are the continuous 2D Gaussian profiles obtained from the sub-pixel estimator.

For the first case of Fig. 4a, there are multiple smooth peaks in the cross-correlation map due to the blur effect. However, the sub-pixel estimator calculates the correct peak location. The second case with image blur in Fig. 4b shows a crown shape with several distinct sharp peaks, similar to the results of Elsinga et al. (2005). The sub-pixel estimator located the peak relatively accurately even for this case. In contrast, in the case of Fig. 5, significant error has occurred. Figure 5a shows that the sub-pixel estimator worked pretty well, but the estimated peak was not correct. Figure 5b shows that the sub-pixel estimator is operating incorrectly, while also incorrectly estimating the peak position to be outside the window.

If problems occur in the cross-correlation map, the error can be minor as in Fig. 4, or major as in Fig. 5. In many cases, it is not easy to clearly categorize the cross-correlation maps into one of the cases described above. If we define displacement error smaller than a threshold of 1 px as nominal error, and anything beyond this an outlier, pixel displacement error for nominal error ($\delta \Delta_i$) and outliers ($\delta \Delta_j$) can be defined below:

![Fig. 4 Cross-correlation map with a multiple smooth peaks resulting in negligible error b multiple sharp peaks resulting in minor error](image-url)
\[
\delta \Delta_i (\text{error}) = \| \Delta_{\text{PIV},i} - \Delta_{\text{real},i} \| \quad \left\{ \begin{array}{l}
\delta \Delta_i \leq 1 \text{px} \\
\delta \Delta_j > 1 \text{px}
\end{array} \right.
\]
\[
\delta \Delta_j (\text{outlier}) = \| \Delta_{\text{PIV},j} - \Delta_{\text{real},j} \|
\]
\[
\langle \delta \Delta \rangle = \frac{1}{N_{\text{error}}} \sum_{i=1}^{N_{\text{error}}} \delta \Delta_i
\]

where \( \| \bullet \| \) indicates the L2 norm operator. For the nominal error case, \( \Delta_{\text{PIV},i} \) and \( \Delta_{\text{real},i} \) represent estimated particle displacement vector by PIV and actual displacement vector using the \( i \)-th image, respectively. For the outlier case, \( \Delta_{\text{PIV},j} \) and \( \Delta_{\text{real},j} \) represent estimated particle displacement vector by PIV and actual displacement vector using the \( j \)-th image, respectively. The mean pixel displacement error (\( \langle \delta \Delta \rangle \)) can then be defined as follows:

\[
\langle \delta \Delta \rangle = \frac{1}{N_{\text{error}}} \sum_{i=1}^{N_{\text{error}}} \delta \Delta_i
\]

where \( N_{\text{error}} \) represents the total number of nominal error cases.

### 2.3 Monte-Carlo method (MCM)

The Monte-Carlo method (MCM) is a good tool for uncertainty evaluation using randomly generated data (Papadopoulos and Yeung 2001), especially when the results are in the form of a complicated multivariable equation (Coleman and Steele 2018). In this study, we utilize MCM to analyze the blur effect on PIV. The variables which are the main parameters that affect PIV error when constructing the synthetic images are shown in Table 1.

| Image size | 32 x 32 | px² |
|------------|---------|-----|
| Particle diameter \( (d_p) \) | Real number | [1, 5] | px |
| Particle number density \( (N_{ppp}) \) | Real number | [0.005, 0.02] | ppp |
| Image noise level \( (\sigma_n) \) | Real number | [0, 0.3] |
| Displacement magnitude \( (\Delta) \) | Real number | [2, 8] | px |
| Blur filter size \( (\kappa_\Delta) \) | Odd number | [3, 9] | px |
| Blur & displacement angle \( (\kappa_\theta) \) | Real number | [0, 2π] | rad |

| Simulated number of images | 100,000 x 5 sets | pairs |
density is represented by the number of particles per pixel (ppp). The directions of motion blur and displacement were set to be the same, and the image generation process was performed using the same value. This is in accordance with the basic PIV premise that there is little or negligible change of the velocity within the PIV window for a short exposure time.

An example using the conditions in Table 1 is depicted in Fig. 6. The data values used in creating Fig. 6 are summarized in Table 2. A total of four images are illustrated: 1) original, 2) original blurred, 3) pre-processed, 4) pre-processed and blurred. The original image corresponds to an image without any processing, and blurred and pre-processed images are processed once accordingly. The pre-processed and blurred image has both image pre-processing and blur treatment. It can be seen that the blur treatment causes particle streaks in one direction. The pre-processing not only reduces background noise, but also increases the particle intensity reduced by blur. Since the image processing is expected to have an impact on the PIV error for the blurred image pairs, an analysis of both the presence and absence of this pre-processing for the MCM is required.

Five simulation sets of 100,000 image pairs were created to quantify the blur effect. By applying the blur filtering in Eq. (3) and pre-processing to these images, four subsets of 25,000 image pairs each were created. In this process, 3 px, 5 px, 7 px, and 9 px blur filters were applied on different images. For each image pair, the exact value of the displacement ($\Delta_{\text{real}}$) is known. Therefore, the blur effect quantification was carried out by comparing the estimated displacement analyzed through the PIV analysis ($\Delta_{\text{PIV}}$) with the known accurate displacement, as shown in Eq. (5).

Before the analysis of the MCM simulation results, we verified that the number of image pairs was sufficient to achieve statistical convergence using the simulation subset mean displacement error $\langle \delta \Delta \rangle$ (Ballio and Guadaginini 2004), as shown in Fig. 7. As the number of image pairs increases, the average value of displacement error for each blur filter size converges at about 5,000, suggesting that the number of images is suitable for performing MCM because there are approximately 25,000 image pairs per filter.

### 2.4 Blur effect estimation

The displacement value difference between PIV analysis of blurred images (i.e., estimated data) and the original displacement from MCM (i.e., reference data) is defined as displacement error ($\delta \Delta$). According to the threshold displacement error, which was defined as 1 px, the various cases can be categorized into either nominal “error” cases or “outlier” cases. The displacement error for the error cases is represented by a relative frequency distribution (RFD), which is illustrated in Fig. 8. There is displacement error even in the absence of blur, which corresponds to inherent PIV error. This is the accuracy limit of the PIV technique itself, and the approximate error of 0.1 px is similar to the uncertainty level described in previous studies (Raffel et al. 2018; Wieneke 2015). As the size of the blur filter increases in Fig. 8, it can be seen that the RFD profile shifts to the right. This trend indicates

---

**Table 2** Example parameters of MCM

| Parameter                           | Value   |
|-------------------------------------|---------|
| Image size                          | 32×32 px² |
| Particle diameter ($d_p$)           | 3.8 px  |
| Particle number density ($N_{ppp}$) | 0.0074 ppp |
| Image noise level ($\sigma_n$)      | 0.17    |
| Displacement magnitude ($\Delta$)   | 6.4 px  |
| Blur filter size ($\kappa_\Delta$)  | 5 px    |
| Blur and displacement angle ($\kappa_\theta$) | 4.47 rad |

---

**Fig. 6** Example of four PIV images based on the given MCM parameters

**Fig. 7** Convergence of mean displacement error based on the number of image pairs

---

© Springer
that displacement errors occur more frequently in cases of relatively large blur filter sizes, as expected. It can also be seen in the right graph that PIV analysis results for images with pre-processing show slightly better results than those of images without pre-processing.

The qualitative trend in Fig. 8 is quantified in Fig. 9. Figure 9, which displays the average $\langle \delta \Delta \rangle$ from 5 sets ($\langle \delta \Delta \rangle$) with blur filter size. As the blur size increases, the average overall displacement error increases. In the absence of blur, an average error of near 0.1 px occurs, whereas in the case of the 9 px blur filter, the mean error value is approximately 0.55 px, regardless of whether pre-processing was conducted or not. The error is large compared to the nominal uncertainty level used in the PIV community, generally considered to be 0.05 – 0.1 px (Raffel et al. 2018; Wieneke 2015). Thus, we conclude that PIV accuracy drops as the blur filter size increases.

![Fig. 8](image.png)

**Fig. 8** RFD of displacement error according to blur filter size for a original case b pre-processed case

![Fig. 9](image.png)

**Fig. 9** Mean value of displacement error according to the blur filter size

![Fig. 10](image.png)

**Fig. 10** Increase in number of outliers with blur filter size

We also compared the frequency of outlier cases (with major error) normalized by the number of total cases, according to the blur filter size. The number of cases where displacement error is greater than 1 px increases with blur filter size, as shown in Fig. 10. While the frequency of outliers in the absence of blur is negligible (0.3–0.4% of $N_{\text{total}}$), the largest blur filter size of 9 px shows a frequency of more than 20% with pre-processing. As the blur filter size grows, the number of outliers increases in a roughly exponential fashion. The uncertainty range based on the 95% confidence interval is also displayed. It should be noted that the uncertainty range is smaller than the symbol size for smaller blur filter sizes.

During the PIV process, it is common to correct outlier cases based on the values of surrounding vectors (Raffel et al. 2018). However, if more than 20% of vectors correspond to outliers, this process can become erroneous. Given that the error caused by outliers and the pixel displacement...
error deviate significantly from nominal PIV uncertainty, it is reasonable to argue that blur effect can cause significant errors in PIV results. Therefore, motion blur treatment in TR-PIV is essential for both cases of standard error and outliers.

3 Deblur using GAN

3.1 Learning procedure

Generative Adversarial Network (GAN) was used to train a generator that outputs a deblurred image from an input image with streaks caused by motion blur. Goodfellow et al. (2014) first proposed the concept of GAN where a generator and discriminator are taught through competition. The learned generator accepts the image without specific treatment as an input, and releases the restored or processed image as an output based on the filter. The discriminator distinguishes which is the original image compared to the restored or processed version obtained from the generator. This puts the generator in a position where it has to deceive the discriminator, and the discriminator is taught to distinguish the fraud that the generator makes. As the discriminator and generator learn competitively, the learning effect is accelerated substantially compared to conventional CNN. Due to these characteristics, the GAN method is known to be suitable for image-to-image translation (Creswell et al. 2018).

We slightly modified the GAN model used in Kupyn et al. (2018) to suit PIV images with motion blur. Deblurring is typically performed on the entire image, but in this study the entire PIV image was not deblurred at once. The image was segmented into 32 px × 32 px sub-windows, and deblurring was performed on these sub-images. Therefore, the layers in the GAN model needed to be modified.

In a previous study, a different learning scheme was applied to PIV images (Grayver and Noir 2020). Instead of GAN, CNN was utilized to conduct particle streak velocimetry, which utilizes the blurry particle streaks within the PIV image to deduce the velocity directly. While this study is different from our study, which improves the performance of PIV analysis by first deblurring the blur motion within the PIV image, it is a good example of how CNN can be used for PIV images.

To learn the GAN generator and discriminator, a large number of blurry and original images are required. However, it is very difficult to capture blurry and original images at the same time in real experiments. Instead, we used synthetic images that can generate large amounts of data freely under well controlled conditions. We produced a total of 1 million original and blurred 32 px × 32 px images. A larger blur filter (3—15px) was applied to the data used in this process compared to the data used in MCM (Sect. 2.3), in order to cover a broader range of cases. In addition, in order to reduce the unnecessary learning cost regarding image noise, we set the noise level of the synthetic images of the training set to zero. Pre-processing was applied to all images corresponding to both the blurred and original image groups. The reason why pre-processing was performed for both groups is to minimize the variables that should be considered in the learning process. Pre-processing increases the homogeneity between images and reduces complexity.

A total of 100 epochs (i.e., number of passes) were attempted using the entire input dataset. The conditions used for learning are listed in Table 3. Figure 11 illustrates the role of the generator and discriminator. The blur image passes through the generator and becomes restored. The restored image then enters the discriminator with the actual original image and is judged by the GAN model as to which image is the real one. In this process, the deviation between the original and restored image is considered perceptual loss, which is used for training the generator. The generator not only uses perceptual loss, but also the discriminator performance score reflected within the WGAN (Wasserstein GAN) loss, which reflects the fact that the generator and the discriminator are competitively adversarial.

| Image size          | 32 × 32 px² |
|---------------------|-------------|
| Particle diameter ($d_p$) | Real number | [1, 6] px |
| Particle number density ($N_{pp}$) | Real number | [0.005, 0.02] ppp |
| Image noise level ($\sigma_n$) | Constant | [0] |
| Blur filter size ($\kappa_s$) | Odd number | [3, 15] px |
| Blur angle ($\kappa_\theta$) | Real number | [0, 2π] rad |
| Training image set size | 1,000,000 images |
| Number of epochs | 100 |

Fig. 11 Schematic diagram of the GAN learning process
3.2 Learning results

The same learning data were passed to the generator for 100 epochs to adjust the weights, and the error value was found to converge, indicating that the generator was constructed well. The generator was well learned because the performance of the discriminator converged to the value of natural selection with a probability of 0.5 (Kupyn et al. 2018). To check the learning results, the data set is largely divided into a test set and validation set. In the case of the test set, 100,000 images, which is 10% of the 1 million images used in the learning procedure, were randomly selected. The validation set used the same number of images, however entirely different from the ones that were previously used for learning. The images previously generated for Sect. 2.3 were used. Figure 12 shows some examples of the deblurring applied to the test set (Fig. 12a) and to the validation set (Fig. 12b). Figure 12a shows deblurred results based on learned data, and it is evident that the deblurred results resemble the original images closer than the blurred ones. The cross-sectional particle intensity distribution which has a flatter profile along the axis of elongation due to blur (as denoted by the arrows) is reconciled toward the original Gaussian profile with deblurring. The results of the generator performance on unstudied data are shown in Fig. 12b, which demonstrates that the deblurring is quite satisfactory. Similarly, for unlearned data, we can see that the original particle intensity profile is nearly recovered with deblurring.

To quantitatively verify the performance of the learning result, Peak Signal-to-Noise Ratio (PSNR) was calculated when the deblur filter was applied to the learned data sets and unstudied data sets, respectively. PSNR is a widely used method to indicate how close the restored image is to the original image, and its definition is given in Eq. (7) (Hradiš et al. 2015; Kupyn et al. 2018; Svoboda et al. 2016; Zhang et al. 2018):

\[
PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right)
\]  

where \( MAX \) is the maximum intensity value (i.e., 255 for this study) and MSE is mean squared error, which is calculated based on the difference in pixel values between the original image \( o(i,j) \) and the image recovered via deblurring or the image with blur \( A(i,j) \), as shown in Eq. (8):

\[
MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [A(i,j) - o(i,j)]^2
\]  

where \( MAX \) is the maximum intensity value (i.e., 255 for this study) and MSE is mean squared error, which is calculated based on the difference in pixel values between the original image \( o(i,j) \) and the image recovered via deblurring or the image with blur \( A(i,j) \), as shown in Eq. (8):

\[
MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [A(i,j) - o(i,j)]^2
\]  

where \( MAX \) is the maximum intensity value (i.e., 255 for this study) and MSE is mean squared error, which is calculated based on the difference in pixel values between the original image \( o(i,j) \) and the image recovered via deblurring or the image with blur \( A(i,j) \), as shown in Eq. (8):

\[
MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [A(i,j) - o(i,j)]^2
\]  

where \( MAX \) is the maximum intensity value (i.e., 255 for this study) and MSE is mean squared error, which is calculated based on the difference in pixel values between the original image \( o(i,j) \) and the image recovered via deblurring or the image with blur \( A(i,j) \), as shown in Eq. (8):
where $m$ and $n$ are the pixel size of images. We used MSE and PSNR to determine if image quality has improved, using the following two equations:

$$\delta_{\text{MSE}} = \text{MSE}_{\text{deblur}} - \text{MSE}_{\text{blur}}$$

(9)

$$\delta_{\text{PSNR}} = \text{PSNR}_{\text{deblur}} - \text{PSNR}_{\text{blur}}$$

(10)

$\delta_{\text{PSNR}}$ increases while $\delta_{\text{MSE}}$ decreases as image quality improves. Based on our results, roughly 83% of the images for learned cases have improved quality, while the rate of improvement is about 70% for unlearned data. This suggests that the generator learned in this study can stably increase image quality, even if the generator encounters new and unlearned data. However, it should be noted that learning itself only improves image quality, and does not directly improve PIV displacement error. Nonetheless, these results suggest that the deblurring generator learned from this study may be effective in improving PIV accuracy. This is discussed in the following section.

3.3 Deblur effect quantification

Using the generator, which restores the blurry image, we quantified the deblur effect for the previously defined validation data set which has not been learned. Figure 13 shows the displacement error RFD for the original no blur case, blur case, and deblur case, similar to Fig. 8. The displacement error value ($\Delta\delta$) was calculated from Eq. (5). The no blur case calculated the error between the PIV analysis using the original image pair without blur and the true value of displacement. The blur case utilized the blurred image pair, and the deblur case used deblurred images which were output from the generator. A total of four graphs are provided according to the blur filter size, and they only depict the standard cases where error values are 1 px or less.

![Fig. 13](image-url) Change in displacement error distribution with deblur treatment, where $\kappa\Delta$ is a 3 px, b 5 px, c 7 px, d 9 px
As the blur filter size increases, the blur case RFD shifts to the right due to larger displacement errors. Deblurring treatment alters the characteristics of the blur case to become more similar to the no blur case. In the case of 3 px and 5 px, which correspond to a relatively small blur filter size, the effect of deblurring moves the RFD almost exactly back to the no blur case. However, the recovery is not as effective for the relatively large blur filter sizes 7 px and 9 px. Nonetheless, it can be seen that the deblur generally shifts the RFD to the left and reduces error compared to the blur case.

For a more detailed description, Fig. 14 illustrates the mean displacement error (with pre-processing), according to blur filter size. As the blur filter size increases, the displacement error of deblur cases increases only a small amount compared to the blur cases, to a value below 0.3 px. For the largest blur filter size, there is almost a factor of 2 reduction in error for the deblurred images. Thus, we have demonstrated that the generator learned through synthetic images can restore streaky PIV images almost to its original state. PIV analysis between deblurred image pairs shows a much lower displacement error compared to the blurred images.

Deblur processing also enables improvements for the outliers. Figure 15 shows the normalized number of outliers with deblur processing for all blur filter sizes. For 3 px and 5 px, the number of outliers was reduced, but the effect was insignificant compared to the baseline case with pre-processing only. However, for the large blur filter sizes, the deblur recovery effect is quite remarkable in reducing the number of outliers which increase almost exponentially for the baseline case. As the blur filter size increases, the increase rate in frequency of outliers is significantly smaller with deblur processing. At the largest blur filter size, the number of outliers is approximately a factor of three smaller.

This result is more dramatic when considering outlier vector correction via neighboring vectors. When the center vector of a 3-by-3 matrix is an outlier, it can be corrected for using the average of the eight neighboring vectors. However, this assumes that these eight vectors are not faulty. If one of the neighboring vectors is inaccurate, the outlier vector will not be properly corrected. If the overall probability of an outlier occurring is more than 20%, when the center vector is an outlier, the probability for at least one more outlier to be included within eight neighboring vectors is roughly 83%. The probability of two or more outliers being included is close to 50%. However, if the probability of outlier occurrence decreases below 6%, these probabilities decrease to under 40 and 8%, respectively. Thus, the reduction in frequency of outlier occurrences will also result in an improvement for PIV outlier correction.

We have hereby shown that the deblurring generator developed in this study can significantly reduce PIV error due to blurry particle images, by decreasing the number of outliers and reducing displacement error.

4 Experimental validation of the deblur filter

4.1 Experimental setup

We conducted an actual PIV experiment of a synthetic jet (similar to Kim et al. 2021) to check the effectiveness of the newly developed deblur method. The experimental setup shown in Fig. 16 was utilized to obtain real PIV particle images with motion blur inside of an acrylic box. The synthetic jet was generated by a woofer speaker attached at the top, and a CW laser illuminated tracer particles from a fluidized bed seeder. The exposure time of a high-speed camera (Phantom v2640) was adjusted to obtain PIV images with
streaks. The nozzle diameter is 20 mm, and the region of interest is about 255 mm directly below the nozzle. Figure 17 is an example of a PIV image where the average streak size is around 7 px, with changing streak directions. A 128 px × 128 px portion of the image is magnified, which illustrates that the particles are elliptical rather than circular.

In order to more accurately determine whether errors caused by motion blur are improved through deburring, simultaneous photography should be performed such that one camera obtains a blurred image, and another camera obtains a sharp image at the same moment and location. However, this type of setup is difficult due to our hardware limitations in PIV lasers and cameras. Another method to evaluate the deblur performance is to measure the velocity field of a flow that is accurately known. Thus, we utilized a water tunnel where the velocity field was known. Blurred images were obtained for the constant velocity flow using a long exposure time, and sharp images through a short exposure time for the high-speed camera. Deblurred images were obtained from the blurred images using the generator.

4.2 Deblur effect on a real PIV image (outlier case)

Since we do not know the true velocity field of the synthetic jet, we were not able to examine the error between measured and true values, and thus instead investigated outliers. This is because the jet flow generated in this research is highly turbulent, and without simultaneous observation, it is impossible to figure out the true velocity values. We divided a 1952 px × 2048 px image into 3904 smaller sub-windows 32 px × 32 px in size. When the PIV analysis was conducted using the deblurred image from the generator, the outliers were remarkably removed. This can be examined via the cross-correlation map example shown in Fig. 18. Figure 18a shows the results of the analysis using the blurry image, and Fig. 18b shows the results after deblurring using the generator. The vector at the center was obtained from the cross-correlation map. The peripheral vectors are also shown to check for outliers. The cross-correlation map is analyzed based on the 32 px × 32 px image, and the estimated peak position is marked with an inverted triangle symbol. As shown in the figure, in the first attempt, motion blur streaks cause multiple peaks to appear in the cross-correlation map. Thus, the wrong peak was selected, causing an erroneous vector. However, as the magnitude of the blur decreases with the deblur filter from the generator, the cross-correlation map is improved, and the correct peak value is found in the second attempt.

Figure 19 displays the instantaneous velocity field results of PIV analysis, conducted using single-pass processing with a square window size of 32 px and 50% window overlap. Figure 20 illustrates a case with dual-pass processing, starting at 64 px and ending at 32 px, both with 50% overlap. When motion blur is severe, the performance enhancement using deblurring is substantial. Outliers were identified if they deviated from a specific displacement range. For both single-pass and dual-pass PIV processing, the number of outliers decreases using our generator. A total of 27 outliers are found for the blur case in Fig. 19, while a total of 6 outliers are found for the deblur case. A total of 20 outliers are found for the blur case in Fig. 20, whereas a total of 9 outliers are found for the deblur case. It can be clearly seen that the deblur filter obtained from GAN learning shows good performance in removing outliers. As noted in Sect. 3.3, outliers from the deblur case can be more easily corrected.
Fig. 18  Example of a real PIV image, vector field, and cross-correlation map for a original blurry case and b deblurred case

Fig. 19 Instantaneous velocity field obtained from a blur case and b deblur case for one pass (32 px × 32 px with 50% overlap) PIV analysis

Fig. 20 Instantaneous velocity field obtained from a blur case and b deblur case for two pass (64 px × 64 px with 50% overlap + 32 px × 32 px with 50% overlap) PIV analysis
through the surrounding vectors because the outliers are likely not adjacent, compared to the blur case.

### 4.3 Deblur effect on a real PIV image (error case)

PIV analysis was performed on images from the water tunnel, and a statistical analysis was conducted on 1000 velocity fields obtained at the center. We used velocity values from a single location within the flow field to examine the statistics, because although the velocity field was constant it might not be perfectly uniform due to variations in laser sheet thickness, image intensity, etc. Three different cases were analyzed: blurred, deblurred, and sharp. Probability density functions (PDF) of the velocity values of each case are shown in Fig. 21. The shape and peak value of the PDF for the blurred case is restored closer to those of the sharp images with image deblurring. Thus, we demonstrate that PIV errors can be reduced with the deblurring filter for a real experiment.

### 5 Conclusion

One of the main causes of time-resolved PIV error is image streaks caused by motion blur. In this study, this blur effect was quantified through synthetic images and the Monte-Carlo method. The PIV error increases with streak size, and the displacement error is 0.2–0.55 px, which is larger than the nominally accepted PIV uncertainty of 0.1 px. Furthermore, the reliability of the PIV analysis could be compromised because the number of outlier vectors is too high. Thus, in order to reduce this blur induced error, a deblurring filter was designed through GAN-based deep learning.

A million blurred images were generated from non-blurry images and were utilized by the learning algorithm to find the correlation between blurry and non-blurry images. The generator (which removes the blur and converts the image into a deblurred image) and the discriminator (which distinguishes the original image from the deblurred image) are mutually competitive. The performance of the discriminator was found to converge at 0.5 probability, which means that the discriminator cannot distinguish which image is real, and proves the fact that the learning was well performed. For the generator, verification procedures were carried out using test sets and validation sets, and improvements in image quality were confirmed. This led to a sharp drop in PIV displacement error and significant reduction of outlier vectors.

The generator, which was developed from synthetic images, was also applied to real PIV images from actual experiments of a synthetic jet and uniform channel flow. We verified that the number of outlier vectors decreases substantially from the synthetic jet, and the error is reduced from the channel flow. Therefore, we believe that our newly developed generator can be used in actual future PIV experiments to obtain more reliable results by reducing the detrimental effects of image blur.

### Acknowledgements

This work was supported under the framework of the international cooperation program managed by the National Research Foundation of Korean (2020K2A9A1A01096358, FY2020), and also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MOE) (No. 2019R1I1A2A01060684). Additionally, this work was supported by the Institute of Advanced Machines and Design and the Institute of Engineering Research at Seoul National University.

### Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

### References

Adrian RJ, Yao CS (1985) Pulsed laser technique application to liquid and gaseous flows and the scattering power of seed materials. Appl Opt 24:44–52

Ballio F, Guadagnini A (2004) Convergence assessment of numerical Monte Carlo simulations in groundwater hydrology. Water Resour Res 40:W04603

Beresh SJ, Neal D, Sciaccitano A (2021) Validation of multi-frame PIV image interrogation algorithms in the spectral domain. In AIAA Scitech 2021 Forum:0019.

Biemond J, Lagendijk RL, Mersereau RM (1990) Iterative methods for image deblurring. Proc IEEE 78:856–883

Bugg JD, Rezkallah KS (1998) An analysis of noise in PIV images. J Visualization 1:217–226

Coleman HW, Steele WG (2018) Experimentation, validation, and uncertainty analysis for engineers. John Wiley & Sons
Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA (2018) Generative adversarial networks: an overview. IEEE Signal Process Mag 35:53–65
Dabiri D (2006) Cross-correlation digital particle image velocimetry-a review. Turchal ABCM Curitiba:155–199.
Elsinga GE, Van Oudheusden BW, Scarano F (2005) The effect of particle image blur on the correlation map and velocity measurement in PIV. Optical Diagn Int Soc Optics and Photonics 1(5880):588010
Elzawawy A (2012) Time resolved particle image velocimetry techniques with continuous wave laser and their application to transient flows. City University of New York.
Goodfellow I, Pouget-Abadie J, Mirza M, Xu B (2014) Generative adversarial nets advances in neural information processing systems. MIT press:2672–2680.
Grayver AV, Noir J (2020) Particle streak velocimetry using ensemble convolutional neural networks. Exp Fluids 61(2):38
Hradiš M, Koteru J, Zemčík P, Šroubek F (2004) The EUROPIV synthetic image generator (SIG) Particle image velocimetry: recent improvements. Springer, Newyork, pp 145–161
Kim M, Lee H, Hwang W (2021) Experimental study on the flow interaction between two synthetic jets emanating from a dual round orifice. Exp Thermal Fluid Sci 126:110400
Kupyn O, Budzan V, Mykhailych M, Mishkin D, Matas J (2018) DeblurGAN: Blind motion deblurring using conditional adversarial networks. Proc IEEE Conf Comput Vision Pattern Recog 1:8183–8192
Lecordier B, Westerweel J (2004) The EUROPIV synthetic image generator (SIG) Particle image velocimetry: recent improvements. Springer, Newyork, pp 145–161
Lee H, Hwang W (2019) Error quantification of 3D homogeneous and isotropic turbulence measurements using 2D PIV. Int J Heat Fluid Flow 78:108431
Mendez MA, Raiola M, Masullo A, Discetti S, Ianoro A, Thunnissen R, Buchlin JM (2017) POD-based background removal for particle image velocimetry. Exp Thermal Fluid Sci 80:181–192
Neeraj PM, Lal SA (2013) Development of a low cost flow visualization and velocimetry apparatus for the investigation of 2D flows. 14th National Conference on Technological Trends:1185–1190.
Nobach H, Honkanen M (2005) Two-dimensional Gaussian regression for sub-pixel displacement estimation in particle image velocimetry or particle position estimation in particle tracking velocimetry. Exp Fluids 38:511–515
Nogueira J, Lecuona A, Rodriguez PA (1997) Data validation, false vectors correction and derived magnitudes calculation on PIV data. Meas Sci Technol 8(12):1493–1501
Papadopoulos CE, Yeung H (2001) Uncertainty estimation and Monte Carlo simulation method. Flow Meas Instrum 12:291–298
Raffel M, Willert CE, Scarano F, Kähler CJ, Wereley ST, Kompenhans J (2018) Particle image velocimetry: a practical guide. Springer
Rossi M, Segura R, Cierpka C, Kähler CJ (2012) On the effect of particle image intensity and image preprocessing on the depth of correlation in micro-PIV. Exp Fluids 52:1063–1075
Scharnowski S, Kähler CJ (2016a) Estimation and optimization of loss-of-pair uncertainties based on PIV correlation functions. Exp Fluids 57:23
Scharnowski S, Kähler CJ (2016b) On the loss-of-correlation due to PIV image noise. Exp Fluids 57:119
Schuon S, Diepold K (2009) Comparison of motion de-blur algorithms and real world deployment. Acta Astronaut 64:1050–1065
Sciaccitano A (2019) Uncertainty quantification in particle image velocimetry. Meas Sci Technol 30:092001
Sciaccitano A, Scarano F (2014) Elimination of PIV light reflections via a temporal high pass filter. Measurement Science and Technology 25:084009.
Sciaccitano A, Scarano F, Wienceke B (2012) Multi-frame pyramid correlation for time-resolved PIV. Exp Fluids 53:1087–1105
Seol DG, Socolofsky SA (2008) Vector post-processing algorithm for phase discrimination of two-phase PIV. Exp Fluids 45:223–239
She W, Zhou G, Guo C, Wu T, Song K (2021) Experimental investigation on the water entry of a bulbous bow based on TR-PIV. Ocean Eng 229:108977
Stanislav M, Okamoto K, Kähler C (2003) Main results of the first international PIV challenge. Meas Sci Technol 14:R63–R89
Svoboda P, Hradiš M, Maršík L, Zemčík P (2016) CNN for license plate motion deblurring. 2016 IEEE International Conference on Image Processing (ICIP). IEEE: 3832–3836.
Thielicke W, Stamhuis E (2014) PIVlab–towards user-friendly, affordable and accurate digital particle image velocimetry in MATLAB. J Open Res Software 2(1):30
Vetel J, Garon A, Pelletier D (2011) Denoising methods for time-resolved PIV measurements. Exp Fluids 51:893–916
Wang S, Chen Y, Liu YZ (2018) Measurement of unsteady flow structures in a low-speed wind tunnel using continuous wave laser-based TR-PIV: near wake behind a circular cylinder. J Visualization 21:73–93
Wieneke B (2015) PIV uncertainty quantification from correlation statistics. Measur Sci Technol 26:074002
Zhang J, Pan J, Ren J, Song Y, Bao L, Lau RW, Yang MH (2018) Dynamic scene deblurring using spatially variant recurrent neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition:2521–2529.
Zhu Y, Yuan H, Zhang C, Lee C (2013) Image-preprocessing method for near-wall particle image velocimetry (PIV) image interrogation with very large in-plane displacement. Meas Sci Technol 24:125302

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Jeong Suk Oh1 · Hoonsang Lee1 · Wontae Hwang1

1 Department of Mechanical Engineering, Seoul National University, Seoul 08826, Korea