

EVALUATIONS OF THE PERFORMANCE OF A CROSS-CORRELATION  
ALGORITHM FOR WIND VELOCITY ESTIMATION USING  
SYNTHETIC BACKSCATTER LIDAR IMAGES  
AND VELOCITY FIELDS

---

A Thesis  
Presented  
to the Faculty of  
California State University, Chico

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Environmental Science

---

by

© Masaki Hamada 2014

Summer 2014

EVALUATIONS OF THE PERFORMANCE OF A CROSS-CORRELATION  
ALGORITHM FOR WIND VELOCITY ESTIMATION USING  
SYNTHETIC BACKSCATTER LIDAR IMAGES  
AND VELOCITY FIELDS

A Thesis

by

Masaki Hamada

Summer 2014

APPROVED BY THE DEAN OF GRADUATE STUDIES  
AND VICE PROVOST FOR RESEARCH:



---

Eun K. Park, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:



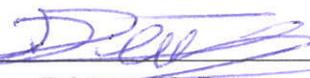
---

Shane D. Mayor, Ph.D., Chair



---

Anna Petrova-Mayor, Ph.D.



---

Pierre Dérian, Ph.D.

## PUBLICATION RIGHTS

No portion of this thesis may be reprinted or reproduced in any manner unacceptable to the usual copyright restrictions without the written permission of the author.

## ACKNOWLEDGMENTS

The research described herein was supported by award 1228464 from the U.S. National Science Foundation's Physical and Dynamic Meteorology Program. I would like to express my appreciation to Dr. Shane D. Mayor, the chair of my thesis advisory committee for taking the time to support this research. Dr. Anna Petrova-Mayor helped me understand the principles of how lasers and atmospheric lidar work. Dr. Pierre Dérian advised me about writing programs that I used for numerical analysis. He and Dr. Petrova-Mayor also proofread the equations in this thesis. Dr. Jakob Mann (Technical University of Denmark) provided his model that generated the turbulent perturbation fields. Christopher Hasek and Christopher Mauzey provided lidar backscatter images from field experiments. I would also like to thank the staff from the following departments at the California State University, Chico, who supported my thesis work: Lidar Group, Department of Geological and Environmental Sciences, Department of Physics, and Graduate Studies. Finally, I would like to express my deep appreciation to my parents Hiroyuki Hamada and Hiroko Hamada, and my sister Masumi Hamada who have supported me during my graduate studies.

## TABLE OF CONTENTS

	PAGE
Publication Rights .....	iii
Acknowledgments .....	iv
Table of Contents .....	v
List of Figures .....	viii
Abstract .....	xii
 CHAPTER	
I. Introduction .....	1
Importance of Wind Measurements .....	1
Methods to Measure the Wind .....	3
Doppler Lidar versus Motion Estimation by Aerosol	
Lidar .....	7
Backscatter Lidar Images .....	12
Wind Velocity Estimation .....	16
Cross-Correlation .....	18
Cross-Correlation Algorithm .....	20
Wind Velocity Fields .....	22
Literature Review .....	23
Purpose of Study .....	26
Hypothesis .....	27
II. Analytical Studies .....	28
Analytical Approaches Using a Gaussian Feature .....	29
Analytical Approaches Using a Rectangular Pulse .....	46
III. Methodology .....	53
Different Types of Wind Velocity Fields .....	54
Effects of Turbulent Intensities .....	71

CHAPTER	PAGE
	Effects of the Edge of a Scan Sector..... 73
IV.	Results ..... 79
	Different Types of Wind Velocity Fields ..... 79
	Effects of Turbulent Intensities ..... 100
	Effects of an Edge of a Scan Sector..... 107
V.	Discussions and Conclusions..... 109
	Synthetic Backscatter Images and Actual Lidar Data ..... 109
	Comparison of Covariance Method and FFT Method ..... 112
	Effects of Zero-Padding..... 115
	Effects of Non-Uniformity..... 118
	Effects of Histogram Equalization ..... 131
	Effects of the Edge of a Scan Sector..... 134
	Features Cut by the Edge of the Interrogation Window ... 136
	Performance of the Cross-Correlation Algorithm ..... 144
	Conclusions ..... 149
	References..... 151
 Appendices	
A.	The Cross-Correlation Function for a Gaussian Feature ..... 161
	Analytical Approaches Using a Gaussian Feature ..... 161
B.	The Cross-Correlation Function for a Rectangular Pulse ..... 184
	Analytical Approaches Using a Rectangular Pulse ..... 184

LIST OF TABLES

TABLE		PAGE
1	The Aerodynamic Roughness Length .....	72
2	Velocity Fields for Turbulent Intensity Test.....	101
3	Turbulent Intensity Test Results .....	101
4	Tests of Non-Uniformity .....	119

## LIST OF FIGURES

FIGURE	PAGE
1. Anemometers .....	5
2. Lidar Scanning Offshore .....	7
3. Doppler Lidar .....	8
4. Doppler Lidar VAD Scan.....	10
5. VAD Scan Plot.....	11
6. Aerosol Lidar .....	12
7. Plan Position Indicator (PPI) Scan.....	14
8. Range Height Indicator (RHI) Scan .....	15
9. Lidar Backscatter Images .....	17
10. 1-D Waveform .....	19
11. Cross-Correlation Function for 1-D Waveform .....	20
12. An Example of Streamlines of the Wind Velocity Field .....	23
13. 1-D Gaussian Feature Moving at Constant Velocity .....	31
14. Cross-Correlation Function for 1-D Gaussian Feature .....	32
15. 2-D Gaussian Feature Moving at Constant Velocity .....	35
16. 1-D Gaussian Feature Moving at Constant Velocity and Diffused Uniformly .....	39
17. Cross-Correlation Function for 1-D Diffused Gaussian Feature.....	41

FIGURE		PAGE
18.	2-D Gaussian Feature Moving at Constant Velocity and Diffused Uniformly .....	44
19.	1-D Rectangular Pulse Moving at Constant Velocity .....	48
20.	Cross-Correlation Function for the 1-D Rectangular Pulse .....	49
21.	2-D Rectangular Pulse Moving at Constant Velocity .....	52
22.	How to Create Synthetic Backscatter Images .....	56
23.	Synthetic Velocity Field .....	58
24.	Synthetic Backscatter Image at Time $t_2$ .....	60
25.	Zero-Padded Synthetic Backscatter Images.....	63
26.	Cross-correlation Function and Displacement Vector .....	65
27.	Multi-Pass Approach .....	67
28.	Synthetic Backscatter Images for Multi-Pass Approach.....	68
29.	Polynomial Fit to Obtain Subpixel Velocity Resolution.....	70
30.	Edge Effect (Lidar Backscatter Images) .....	75
31.	Edge Effect (Zoomed Lidar Backscatter Images).....	76
32.	An Example of Edge Effect .....	77
33.	Test of Edge Effect.....	78
34.	Result for Uniform Flow.....	81
35.	Result for Uniform Flow with Turbulent Perturbations .....	83
36.	Result for Convergent Flow.....	85
37.	Result for Convergent Flow with Turbulent Perturbations .....	87
38.	Result for Divergent Flow.....	89

FIGURE		PAGE
39.	Result for Divergent Flow with Turbulent Perturbations .....	91
40.	Result for Rotational Flow.....	93
41.	Result for Rotational Flow with Turbulent Perturbations.....	95
42.	Result for Shear Flow .....	97
43.	Result for Shear Flow with Turbulent Perturbations .....	99
44.	Velocity Fields Generated by Changing Roughness Lengths .....	103
45.	Results for Changing Roughness Lengths.....	104
46.	Turbulent Intensity Test Results .....	105
47.	Turbulent Intensity Test Results .....	106
48.	Result for Testing Edge Effect .....	108
49.	Comparison of REAL lidar backscatter image and synthetic backscatter image .....	111
50.	Comparison of Covariance Method and Fast Fourier Transform Method .....	114
51.	Effect of Zero-Padding .....	117
52.	Non-Uniformity Test 1 .....	121
53.	Non-Uniformity Test 2 .....	122
54.	Non-Uniformity Test 3 .....	124
55.	Non-Uniformity Test 4 .....	125
56.	Non-Uniformity Test 5 .....	128
57.	Non-Uniformity Test 6 .....	129
58.	Non-Uniformity Test 7 .....	130

FIGURE		PAGE
59.	Effect of Histogram Equalization.....	133
60.	Example of Gaussian Feature Cut by a Edge Line .....	135
61.	Feature Entering the Interrogation Window.....	137
62.	Feature Leaving the Interrogation Window.....	139
63.	Tukey Windows.....	141
64.	Feature Entering the Interrogation Window with the Tukey Window .....	142
65.	Feature Leaving the Interrogation Window with the Tukey Window	143
66.	Maximized the Perfomance of the Cross-Correlation Algorithm ...	146
67.	False Apparent Motion of a Spherical Feature.....	148
68.	Limits of Integration in the 2-D Cartesian and Polar Coordinates	165
69.	Contour Integration.....	185

## ABSTRACT

# EVALUATIONS OF THE PERFORMANCE OF A CROSS-CORRELATION ALGORITHM FOR WIND VELOCITY ESTIMATION USING SYNTHETIC BACKSCATTER LIDAR IMAGES AND VELOCITY FIELDS

by

© Masaki Hamada 2014

Master of Science in Environmental Science

California State University, Chico

Summer 2014

The performance of a cross-correlation algorithm for remote wind velocity estimation by atmospheric lidar is evaluated by using synthetic aerosol backscatter images and synthetic wind velocity fields. The results of applying the cross-correlation algorithm to these synthetic images and flow fields were compared to the mean velocities of the synthetic flow fields. To mimic the atmospheric backscatter, a gently varying aerosol background was generated from smoothed random noise and the superposition of Gaussian puffs. Two-component vector flow fields with spatial variability were generated by analytic formulations for various flows such as convergence, divergence, rotation, and shear. Small scale turbulent perturbations were generated by use of a spectral tensor model of atmospheric surface layer turbulence. The pixels of synthetic backscatter from the first image were displaced by the corresponding vectors of the synthetic flow fields and a second image was generated by bicubic interpolation.

The cross-correlation algorithm results in one displacement vector that maximizes the correlation of two consecutive backscatter images. The results show that the cross-correlation algorithm produces perfect displacement vectors only when the velocity field is perfectly uniform and no aerosol features enter or leave the interrogation window between consecutive frames. Any spatial variation in the velocity field may result in an error and the magnitude of the error increases as the non-uniformity of the flow field increases.

The results show that the magnitudes of the resultant displacement vectors tend to be slightly lower than the actual ones from the given velocity fields. There are two possible causes of underestimation. The first is likely attributed to the non-uniformity of the velocity field within the interrogation window. In this case, aerosol features with lower velocity tend to appear in both frames, while those with higher velocity tend to move out of the interrogation window. Reducing the size of the interrogation window may reduce this error. The second is likely attributed to aerosol features cut by the image edges. In this case, a larger interrogation window is permissible if the edges of the images are tapered. The results also show that both the area and the intensity of aerosol features influence the performance of the cross-correlation algorithm. This observation can be reduced by applying histogram equalization. However, the histogram equalization tends to broaden the cross-correlation function and increases the ambiguity of the peak location.

According to this study, the performance of the cross-correlation algorithm is maximized by: (1) using zero-padded images, (2) tapering the image edges, (3) applying a multi-pass approach, and (4) fitting curves to the cross-correlation function peak for subpixel resolution of the displacement vector. For experiments with 100 pairs of synthetic backscatter images applied to synthetic

velocity fields of relatively low turbulent intensities, the cross-correlation algorithm results agree with the actual mean velocity within 2% with these options. However, the underestimation of the displacement vector cannot be completely eliminated.

## CHAPTER I

### INTRODUCTION

In this chapter, the importance of wind measurements is briefly discussed and several techniques to measure the wind velocity are shown. These techniques are classified into two types, in situ (in position), that is the measurement based on direct contact, and remote sensing. The cup anemometer and the sonic anemometer are two examples of instruments for in situ measurement of the wind. On the other hand, the Doppler lidar and the aerosol lidar are two examples of instruments for remote sensing of the wind. Both the Doppler lidar and the aerosol lidar can be used to estimate horizontal wind velocity vectors. From these two types of lidar, this thesis focuses on the use of the cross-correlation algorithm to estimate the wind field from aerosol lidars. A review of literature in this chapter lists examples of the cross-correlation algorithm previously applied to estimate the wind velocity. Finally, a statement of problems and hypothesis associated with the cross-correlation algorithm are described at the end of this chapter.

#### Importance of Wind Measurements

Wind is created by horizontal atmospheric pressure gradients,<sup>1</sup> but landscape, vegetation, and terrestrial features such as houses and buildings strongly influence the wind velocity near the surface of the Earth. In general, surface vegetation, features, and terrain reduce the wind velocity at the surface because of

<sup>1</sup>A physical quantity that describes which direction and at what rate the pressure changes around a particular location.

the frictional force they exert on the flow (Stull, 1988). On the other hand, when the wind is blown over a ridge that lies transverse to the wind flow, the air streams may be squeezed and accelerated, which results in higher wind velocity on elevated hilltops. Similarly, convection may influence the mean wind velocity near the surface.

Wind is a renewable energy resource, and the measurement of the wind velocity is crucial to assess wind energy potential in a given region. The power available from the wind is proportional to the density of the air, the area of the turbine rotor disk, and the cube of the wind speed. The wind power  $P$  is expressed as

$$P = \frac{1}{2}C\rho Av^3, \quad (1)$$

where  $C$  is the power coefficient,  $\rho$  the density of the air,  $A$  the area of the turbine rotor disk, and  $v$  the wind speed (Burton et al., 2011). These factors encourage industry to develop larger wind turbines at higher altitudes (where the frictional effect of the surface is reduced and the average wind speed higher) and with larger diameter rotors. Modern wind turbines, capable of producing 5 MW of electrical power, have hubs typically at 100 m above the surface, and with rotor diameters approaching 100 m.

The ocean is relatively flat compared to land, so the frictional effects of the ocean surface on the wind are relatively small and the wind speeds are generally larger over the ocean than over the land. Significant interest exists in installing wind farms (arrays of wind turbines) offshore. However, offshore wind farm developers must demonstrate that adequate wind resources exist in order to acquire the financing needed. Observations of the wind over long periods are a component of wind resource assessment (Bailey et al., 2012).

In addition to holding value for use in wind resource assessment, lidar observation of the wind field is likely to have value in very short-term prediction of the wind. Forecasts of the wind, up to a few minutes in advance, could be used to optimize the production of wind energy and protect turbines from damage. Finally, in addition to the above applications in wind energy, remote wind measurements may be of value in a variety of other fields. Those include transportation (especially aviation safety near airports); wildfire suppression; prediction of atmospheric pollution dispersion; and meteorological research.

## Methods to Measure the Wind

### Anemometer

An anemometer is a device used to measure the wind speed. The first anemometer was a swinging-plate instrument described and illustrated by Leon Battista Alberti in 1450 and the shape and structure of the anemometer has evolved to measure the wind speed accurately (Middleton, 1969). Among several types of anemometers, two types, a cup anemometer and a sonic anemometer, are often used to measure the wind speed today. A cup anemometer consists of hemispherical cups mounted on the ends of horizontal arms mounted at equal angles to each other on a vertical shaft, as shown in Figure 1a. When the wind blows from any horizontal direction, torque is produced, and the vertical shaft turns. For this device, the wind velocity is proportional to the angular speed of the shaft. Thus, one can measure the average wind speed by counting the number of turns of the shaft over a given time period. Some advantages of this measurement are low cost and generally good accuracy (Bailey et al., 2012).

Sonic anemometers use ultrasonic pulses to measure the wind velocity. Figure 1b represents the sonic anemometer. When an ultrasonic pulse is emitted

from a transducer, it travels to another transducer. The time required for a given pulse to travel across a pair of transducers depends on the wind velocity. For example, if the wind blows in the opposite direction of the pulse's velocity, it takes longer time than in the absence of wind. The wind velocity can be measured by the combination of measurements of the time to travel between pairs of transducers. The advantage of this measurement is that sonic anemometers are "fast response" and therefore can measure turbulent fluctuations of the wind velocity which is common in the planetary boundary layer.<sup>2</sup>

The wind measurement using three cup anemometers, as shown in Figure 1a, is the most popular method, and is currently used as the industry standard for wind resource assessment studies (Bailey et al., 1997). However, cup or sonic anemometers give only the wind speed at a specific point in space. Moreover, this measurement requires a stationary platform such as a tower. It is very difficult to erect towers over the ocean, so in situ measurement is challenging to deploy for offshore winds.

<sup>2</sup>The bottom layer of the troposphere that is in contact with the surface, and responds rapidly to the surface.



(a) Cup anemometer.



(b) Sonic anemometer (CSAT3).

Figure 1. (a) An image of a cup anemometer (Met-One speed sensor). The wind speed is measured by the rate of rotation of cups. (b) An image of a sonic anemometer (CSAT3: three-dimensional sonic anemometer). The wind velocity can be measured by the combination of measurements of time of travel of ultra sonic pulses from pairs of transducers. Source: Images from Campbell Scientific, Inc.: <http://www.campbellsci.com/>.

## Lidar

Lidar is an acronym for LIght Detection And Ranging. Lidar is an active remote sensing technology that illuminates distant targets with radiation, and is similar in principle to radar (RAdio Detection And Ranging). The main difference is that lidar uses electromagnetic radiation of wavelength (200 nm - 10  $\mu\text{m}$ ) and is much shorter than that of radar (1 mm - 10 cm). Since shorter wavelengths of radiation allow the detection of smaller scatterers in the atmosphere, lidar systems are more suitable for observing the clear atmosphere than radar systems. Therefore, lidars can be used to measure the wind velocity via the scattering of aerosol particles.<sup>3</sup> This measurement is attractive for offshore winds since it may be possible to measure the offshore wind velocity remotely from the coast. Figure 2 is an artist's rendition of a lidar system scanning from the coast.

Atmospheric lidars that are capable of observing the wind can be divided into three broad classes (Baker and Coauthors, 2013). First, *coherent Doppler* lidars employ heterodyne detection to measure the frequency shift of the backscattered radiation (Grund et al., 2001, Henderson et al., 1991, Pearson et al., 2009, Post and Cupp, 1990). Second, *direct detection Doppler* lidars employ various types of interferometers to measure the frequency shift of the backscattered radiation (Gentry et al., 2000, Gentry and Korb, 1994, Grund and Tucker, 2011, McGill et al., 1997). The third method to measure the wind is by the use of motion estimation algorithms applied to the images produced by direct detection non-Doppler lidars (Eloranta et al., 1975, Kolev et al., 1988, Shimizu et al., 1981). Because coherent detection Doppler lidars are commercially available and most commonly used in lower atmospheric wind sensing, the next section describes how

<sup>3</sup>Colloid of fine solid particles or liquid droplets suspended in a fluid.

coherent Doppler lidars are used to observe vertical profiles of two-component horizontal wind vectors.

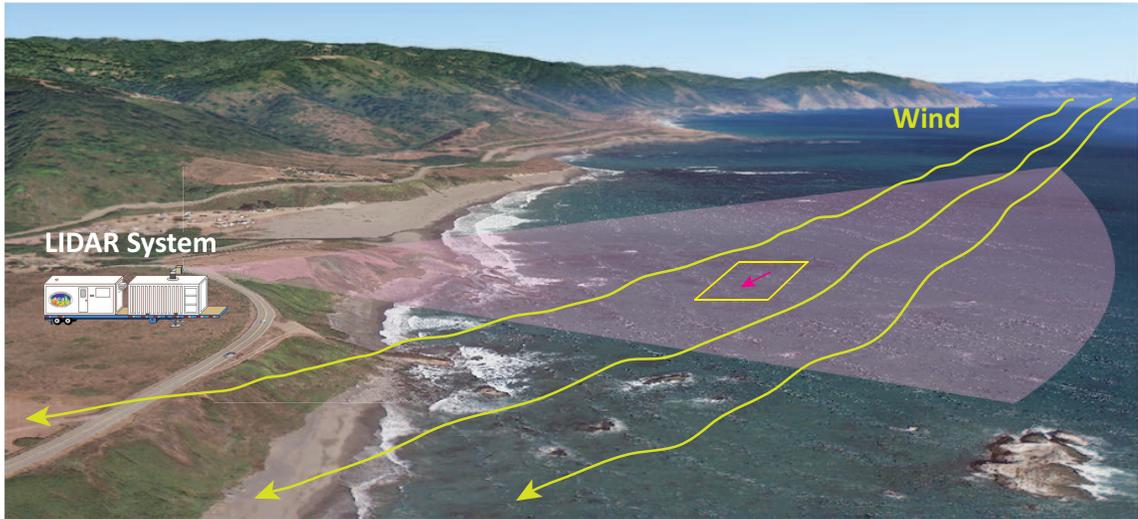


Figure 2. An artist's rendition of a lidar system scanning from the coast. Note the square block on the scan sector. This square block represents an example of an "interrogation window" used by the cross-correlation algorithm.

### Doppler Lidar versus Motion Estimation by Aerosol Lidar

#### Doppler Lidar

Doppler lidar is a remote sensing technology used to measure the wind velocity. Figure 3 shows an artist's rendition of a commercially available compact Doppler lidar. A Doppler lidar emits pulses of electromagnetic radiation into the atmosphere and measures the radial (or line-of-sight) component of the wind velocity via the Doppler shift of frequency of backscattered radiation by the motion of aerosol particles. When the frequency of the backscattered radiation is higher than that of the transmitted radiation, the aerosol particles are approaching the

Doppler lidar. On the other hand, when the frequency of the backscattered radiation is lower than that of the transmitted radiation, the aerosol particles are moving away from the Doppler lidar. To obtain the distance between the Doppler lidar and the pulse volume as it travels, the time between emitting and receiving radiation is measured. From the Doppler shift of the backscattered radiation, the distance between the Doppler lidar and aerosol particles, and the fact that aerosol particles move with the wind, the radial component of the wind velocity (either approaching or receding), at remote locations, can be estimated.

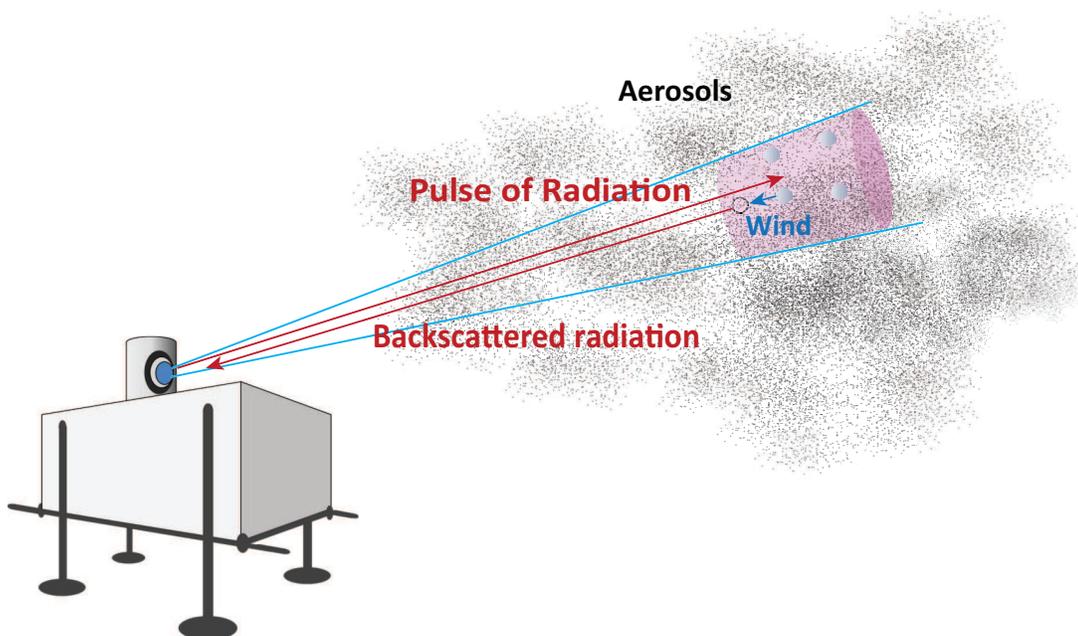


Figure 3. An artist's rendition of a commercially available, compact, heterodyne Doppler lidar, and a single pulse at an instant in time. The pulse propagates through the atmosphere at the speed of light.

To estimate the horizontal wind velocity vector using a Doppler lidar, a conic scan and curve fitting must be used. The velocity azimuth display (VAD)

method results in a vertical profile of horizontal wind vectors above the Doppler lidar (Köpp et al., 1984). It is a “profiling” method. Figure 4 shows a special case of VAD scans. The x-, y-, and z-directions are defined as eastward, northward, and upward, respectively. Here, the wind velocity is in the positive x-direction (eastward), and a Doppler lidar scans from the azimuthal angle  $\phi = 0$  to  $\phi = 2\pi$  while the the polar angle  $\theta$  is held constant. In this case, the Doppler lidar scans the atmosphere in the shape of a cone. At point A, the radial component of the wind velocity is positive and maximum, then it decreases to zero (at point B), minimum (at point C), and increases to zero (at point D), and returns to the maximum value (at point A). Figure 5 shows the radial component of the wind velocity  $v_r$  versus the azimuthal angle  $\phi$  for this special case of VAD scans which demonstrates that  $v_r$  changes sinusoidally as  $\phi$  increases. In this special case, the plot is the cosine function when the phase angle equals zero. If the wind has both the north and east components, the phase angle becomes a non-zero value. Doppler lidar has been used to measure wind profiles, and experimentally validated (Mann et al., 2010, Sathe and Mann, 2012, Sathe et al., 2011). The advantages of this measurement are the ability to measure the wind velocity remotely, and the high reliability of the technique (Reitebuch et al., 2001). The disadvantages of this measurement are that the method assumes that the wind velocity is uniform throughout the region swept by the Doppler lidar, and that it takes time to sweep out the cone. Also, the method is not always accurate when the wind velocity is not uniform in the region inside of the cone or changes in the time interval for one complete cycle of the VAD scan. It is possible to scan a Doppler lidar horizontally as in Figure 2, and obtain radial velocity fields, but the full description of the wind velocity field is not possible from a single Doppler lidar. According to Newsom

et al. (2005), dual-Doppler is possible, but using two Doppler lidars increases the cost and complexity.

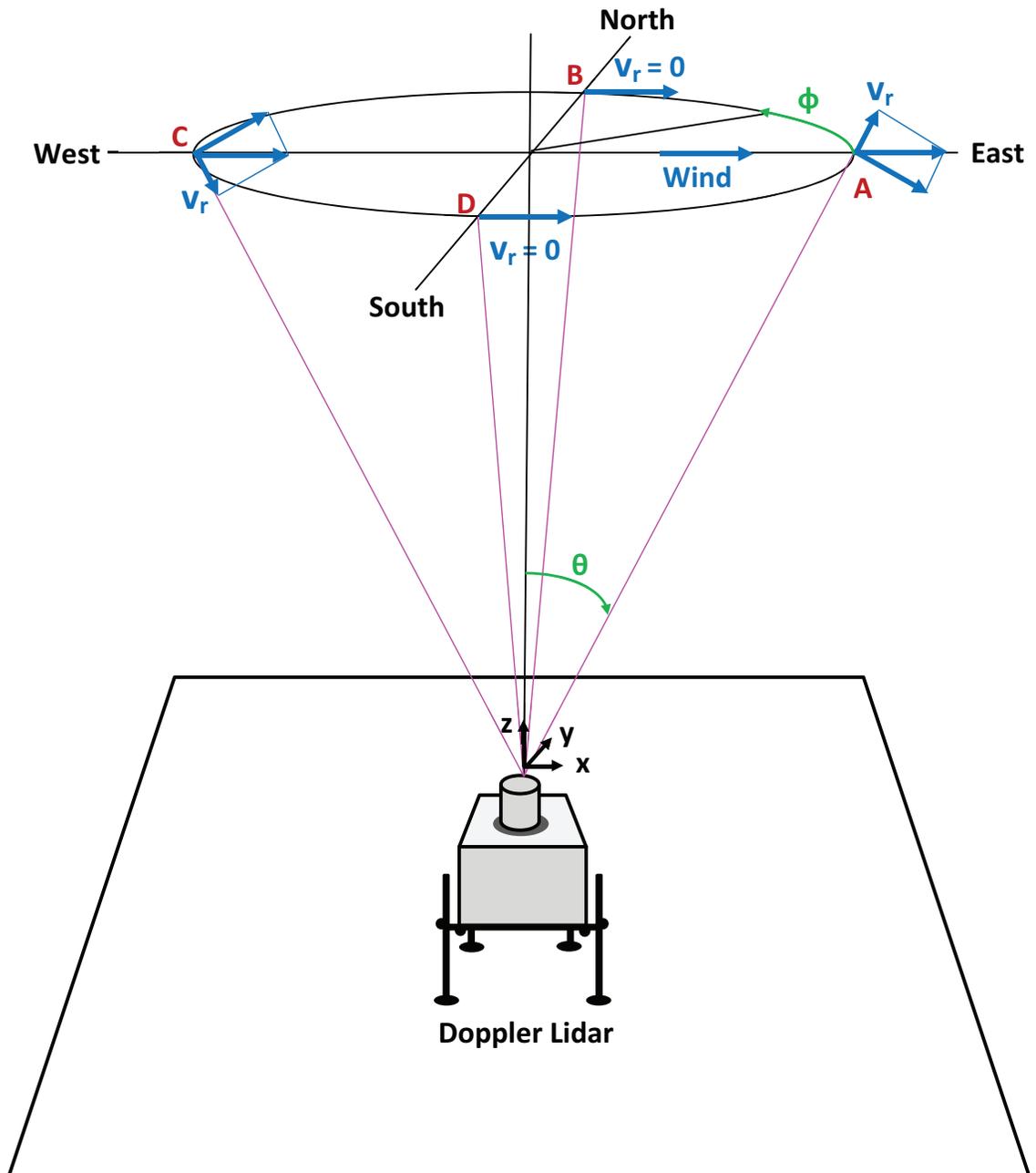


Figure 4. The Doppler lidar system used to measure 2-components of wind velocity (VAD scans).

### Radial Component of the Wind Velocity $v_r$

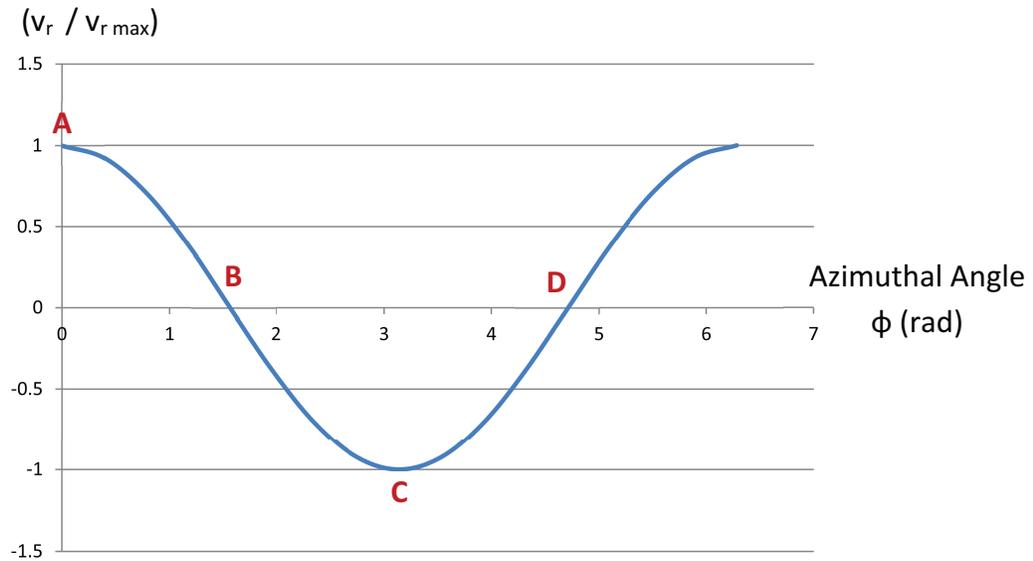


Figure 5. The radial component of the wind velocity  $v_r$  versus the azimuthal angle  $\phi$  for VAD scan of Doppler lidar.

### Motion Estimation by Aerosol Lidar

Aerosol lidar is another remote sensing technology that may be used to measure the wind velocity. Figure 6 represents the aerosol lidar, the REAL, which stands for Raman-shifted Eye-safe Aerosol Lidar. REAL is an elastic backscatter lidar operating at  $1.54\text{-}\mu\text{m}$  wavelength of radiation (Mayor and Spuler, 2004, Mayor et al., 2007, Spuler and Mayor, 2005, 2007). Similar to the Doppler lidar, this lidar system emits pulses of electromagnetic radiation into the atmosphere, and collects the backscattered radiation by aerosols. However, this lidar system measures the wind velocity by applying motion estimation algorithms to consecutive pairs of backscatter images (Mayor et al., 2012). The advantage of this approach is the ability to obtain two components of the wind velocity over an area from a single

lidar system. The following sections describe how to measure the wind using this lidar system.

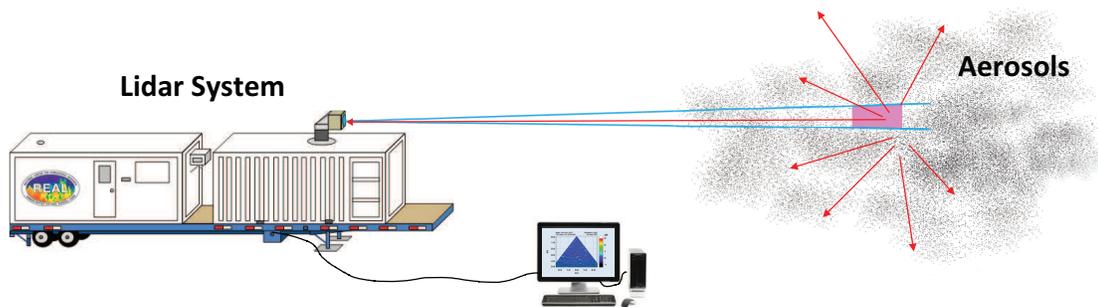


Figure 6. An artist's rendition of the Raman-Shifted Eye-Safe Aerosol Lidar (REAL) at an instant in time after a laser pulse is transmitted into the atmosphere.

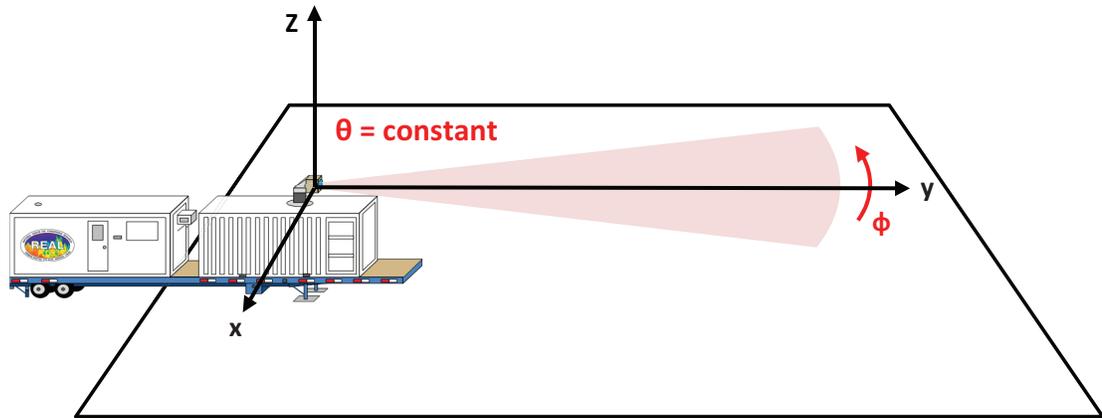
### Backscatter Lidar Images

The lidar system observes the atmosphere as follows. First, the lidar system emits short pulses of radiation into the atmosphere. For example, the lidar system described by Mayor et al. (2007) emits a 170-mJ pulse of 6-ns pulse duration every 0.1 s. Next, the pulse of radiation is scattered in all directions by aerosol particles, a colloidal systems of solid or liquid particles suspended in the atmosphere, as the pulse traverses the atmosphere. (The size of an aerosol particle ranges approximately from  $0.1 \mu\text{m}$  to  $100 \mu\text{m}$ .) As the pulse propagates through the atmosphere, a small portion of the radiation is scattered back to the lidar system and focused on to a photodetector. Backscattered radiation focused on the photodetector is converted to an electrical current by the photodetector and amplified by a transimpedance amplifier. The resulting analog signal is converted to a digital signal by a high-speed digitizer.

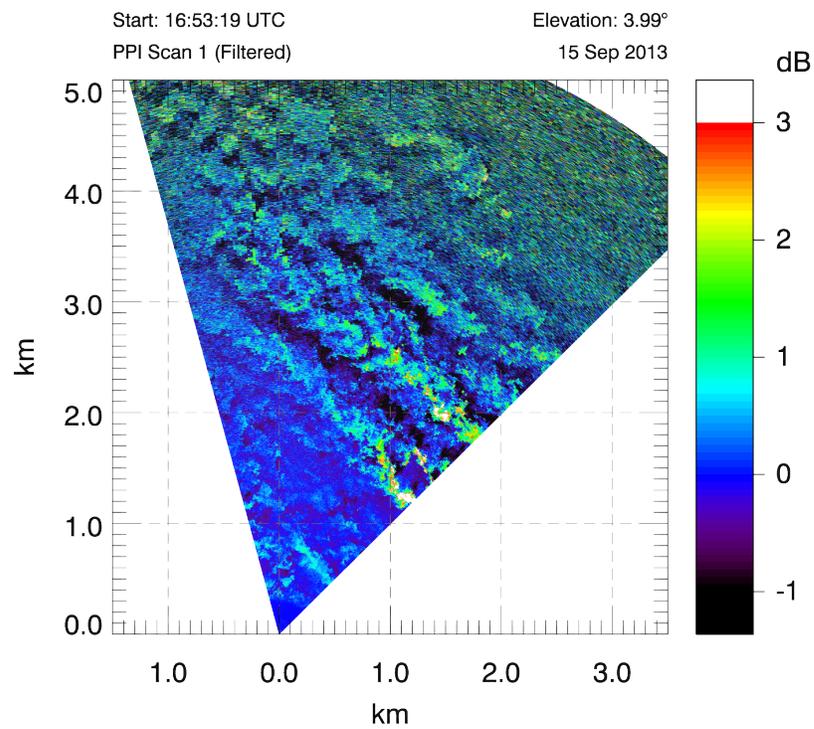
The scanner (or beam steering unit) directs subsequent pulses in other azimuth or elevation angles. Finally, from the collection of returns at different look angles, an image of backscatter intensities due to aerosols may be created. Since each aerosol particle is small compared to the pulse volume, the lidar cannot distinguish individual aerosol particles. However, it can measure what is a proxy to relative aerosol concentration via backscatter intensities and detect significant macroscopic aerosol features in the images.

In practice the REAL lidar backscatter signal intensity data is converted to relative aerosol backscatter intensity by multiplying each element of returned waveform by the range squared. In addition, the range-corrected relative backscatter intensity is expressed in decibels (dB). Finally, high-pass median filtering is applied to remove large-scale features that are not likely to be advected by the local wind. These features, for example, include attenuation, instrumental artifacts, and large meteorological features such as mesoscale fronts.

There are two types of lidar scans, plan position indicator (PPI) and range height indicator (RHI). Let  $\theta$  stand for the polar angle (angle from the vertical  $z$ -axis) and  $\phi$  for the azimuthal angle. Then, the PPI scan sweeps in the direction of increasing  $\phi$  while  $\theta$  is held constant, as shown in Figure 7a. Then one interpolates the polar data to a Cartesian grid to simplify image processing and motion estimation. The result is that one obtains the horizontal cross-section of the atmosphere, as shown in Figure 7b. When the RHI scan sweeps in the direction of increasing  $\theta$  while  $\phi$  is held constant, a vertical cross-section of the atmosphere is obtained, as shown in Figure 8b. This image shows the planetary boundary layer is about 800 m deep. Thus, the lidar system can be used to observe the atmosphere in two dimensions via backscattered radiation by aerosols.

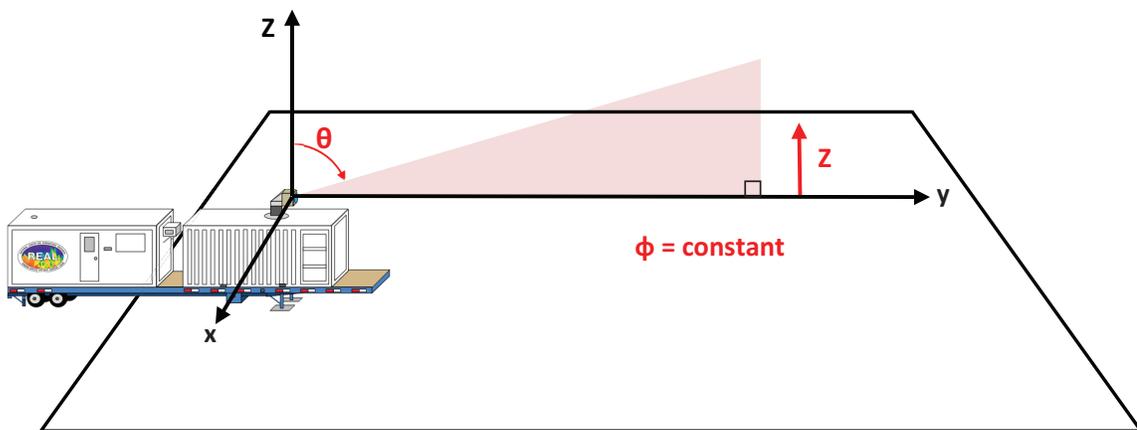


(a) The Plan Position Indicator (PPI) scan by the REAL.

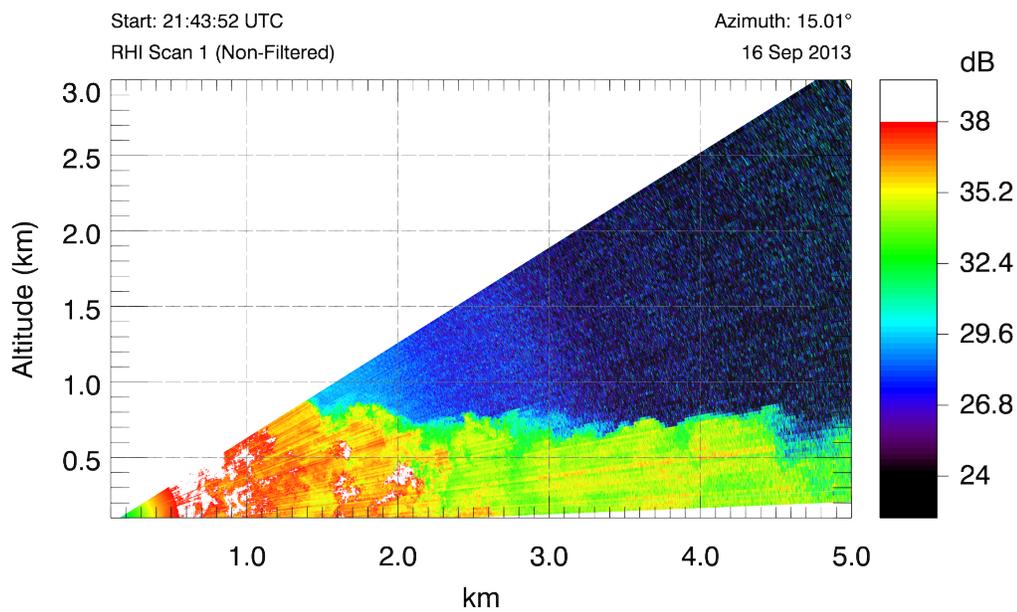


(b) An example of PPI scan lidar image.

Figure 7. Plan Position Indicator (PPI) scan.



(a) The Range Height Indicator (RHI) scan by the REAL.



(b) An example of RHI scan lidar image.

Figure 8. Range Height Indicator (RHI) scan.

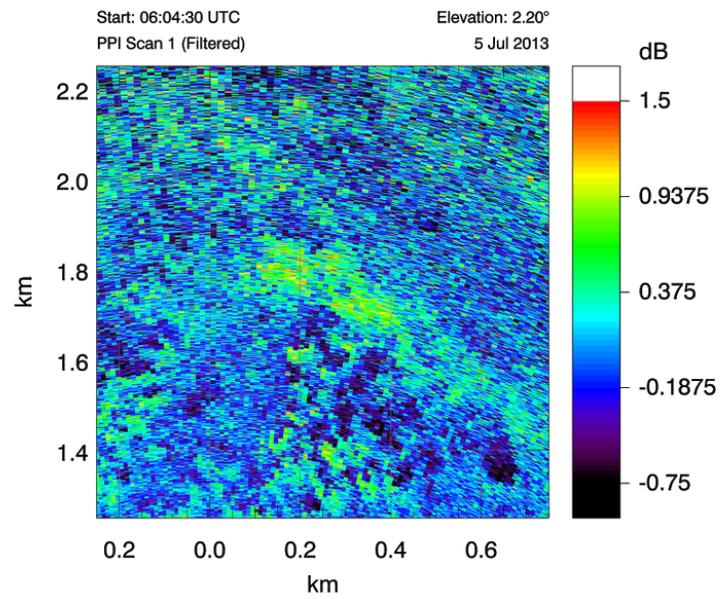
### Wind Velocity Estimation

The wind velocity is estimated by using two consecutive backscatter lidar images. The primary assumption is that aerosol features are advected by the wind. In this case, the wind velocity in a particular square region, called a block, should be approximately equal to the velocity of the aerosol features in the region. To find the velocity of aerosol features from lidar backscatter images, two consecutive images are taken to find the displacement of the aerosol features  $\Delta\vec{r}$ . Then, the velocity  $\vec{v}$  of the aerosol features can be calculated by

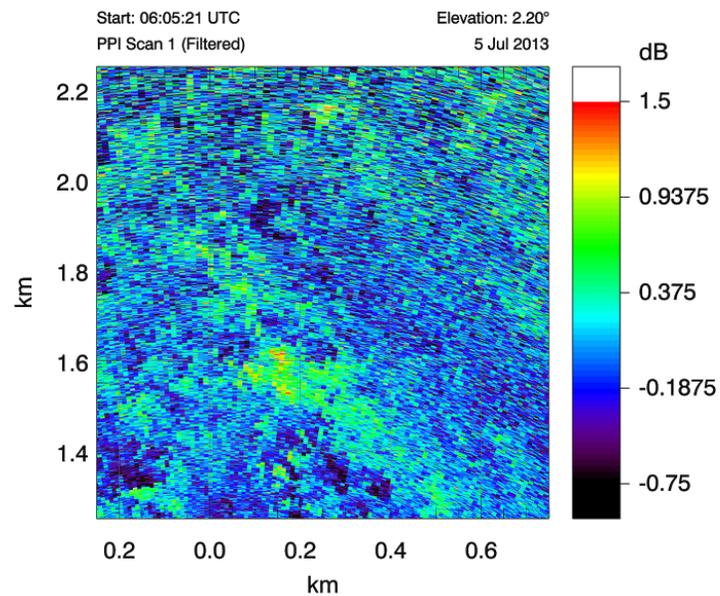
$$\vec{v} = \frac{\Delta\vec{r}}{\Delta t}, \quad (2)$$

where  $\Delta t$  is the time between two consecutive lidar scans.  $\Delta t$  depends on how fast the lidar system scans the atmosphere, but for the REAL it is about 15 s for a  $60^\circ$  sector scan. The lidar system's scan speed  $\Delta t$  is constant. However, it is challenging to get displacement  $\Delta\vec{r}$  between the two consecutive images since aerosol features tend to diffuse and change shape and brightness as they move.

Figure 9 shows two real lidar backscatter images separated by  $\Delta t = 51$  s. One can see that aerosol features have moved to the south and it is the job of the motion estimation algorithm to estimate the displacement of these features in the time interval  $\Delta t$  between the scans. In this case, motion estimation methods may be applied to derive the displacement  $\Delta\vec{r}$  and one motion estimation algorithm is the cross-correlation algorithm (Eloranta et al., 1975, Mayor et al., 2012).



(a) Lidar backscatter image at  $t_1$ .



(b) Lidar backscatter image at  $t_2$ .

Figure 9. Two lidar backscatter images separated by the time interval  $\Delta t = t_2 - t_1 = 51$  s. They were collected at the California State University, Chico, University Farm.

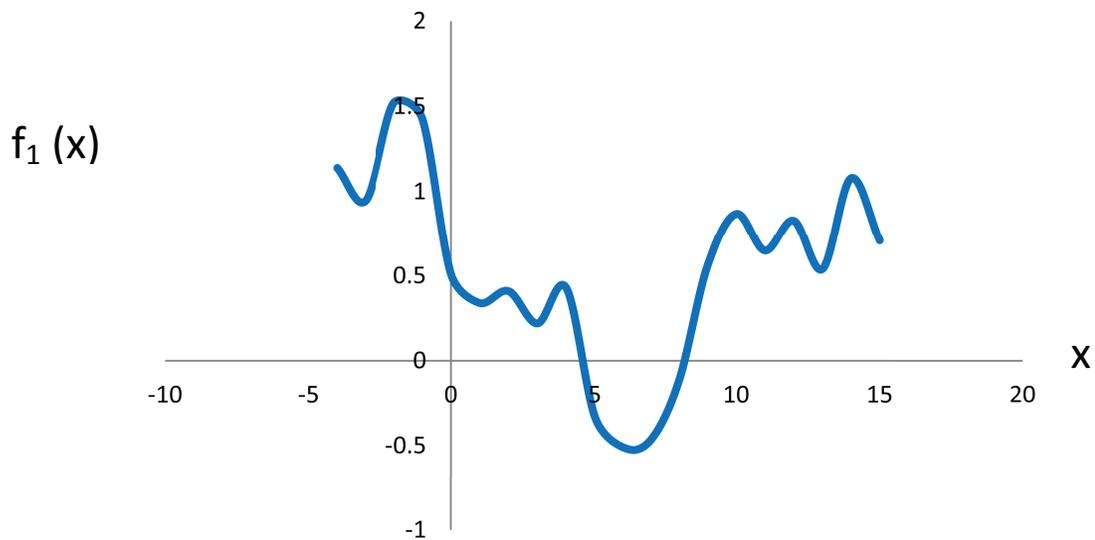
### Cross-Correlation

The cross-correlation function provides a measure of the similarity of two waveforms as a function of lag<sup>4</sup> applied to one of the waveforms. The normalized cross-correlation function,  $r_x$ , for two waveforms  $f_1$  and  $f_2$  is defined as

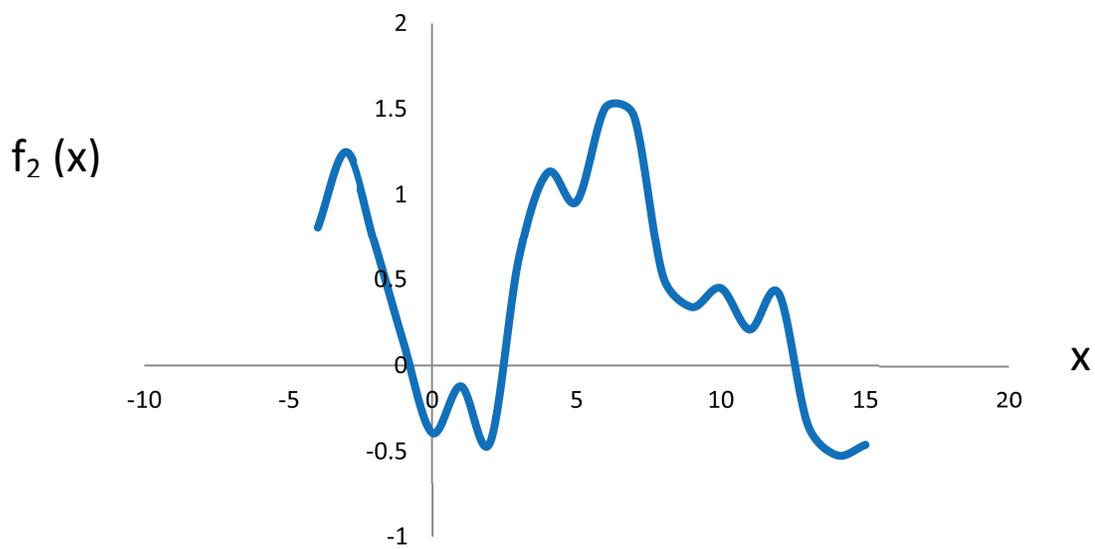
$$r_x = \frac{COV_{1,2}}{S_1 S_2}, \quad (3)$$

where  $COV_{1,2}$  is the covariance of the overlapped portions of  $f_1$  and  $f_2$ ,  $S_1$  is the standard deviation of  $f_1$ , and  $S_2$  is the standard deviation of  $f_2$  (Davis and Sampson, 2002). Figures 10a and 10b show an example of 1-dimensional (1-D) waveforms  $f_1$  and  $f_2$  respectively, and Figure 11 shows the resulting cross-correlation function,  $r_{x_{lag}}$ , applied to these waveforms, and plotted as a function of  $x_{lag} = -x$ . From Figure 11, the cross-correlation  $r_{x_{lag}}$  is maximum when the lag,  $x_{lag} = 8$ . In this case, the overlapping portions of two waveforms are most similar when the second waveform is shifted 8 units with respect to the first. If the second waveform were moved with respect to the first, the lag,  $x_{lag} = 8$ , represents the displacement of the second waveform that maximizes the correlation of the two waveforms. Thus, cross-correlation can be used to estimate the displacement of the second waveform, and this principle can be applied to estimate the displacement of aerosol features in the time interval  $\Delta t$  between the two consecutive backscatter images.

<sup>4</sup>The negative of a given coordinate. For example,  $x_{lag} = -x$ .



(a) Arbitrary 1-D waveform at time  $t_1$ .



(b) Arbitrary 1-D waveform at time  $t_2$ .

Figure 10. An example of 1-D waveforms. At time  $t_1$ , the waveform is expressed as  $f_1(x)$ . The waveform changes in the time interval  $\Delta t = t_2 - t_1$ . At time  $t_2$ , the waveform is expressed as  $f_2(x)$ .

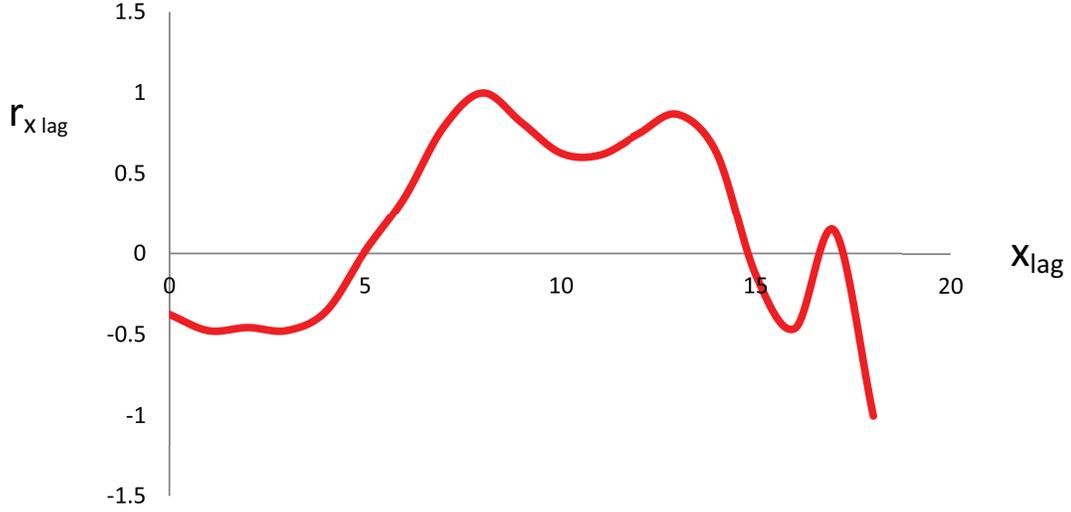


Figure 11. The cross-correlation function for the 1-D waveforms shown in Figure 10. The peak location of the cross-correlation function  $r_{x_{lag}}$  represents the displacement of the waveform  $f_1(x)$  that maximizes the correlation of the two waveforms  $f_1(x)$  and  $f_2(x)$ .

### Cross-Correlation Algorithm

The cross-correlation algorithm is a numerical procedure to determine the displacement of features between two consecutive frames. The displacement is determined by finding the location of the peak of the cross-correlation function (Eloranta et al., 1975). For continuous functions  $f_1(x)$  and  $f_2(x)$ , the cross-correlation function,  $r_x$ , is defined as

$$r_x = \int_{-\infty}^{\infty} f_1^*(x) f_2(x - x_{lag}) dx, \quad (4)$$

where  $f_1^*(x)$  is the complex conjugate of  $f_1(x)$ . According to Telford et al. (1990), the cross-correlation function of  $f_1(x)$  and  $f_2(x)$  is equivalent to the convolution of  $f_1(x)$  with  $f_2(-x)$ .

The cross-correlation function  $r_x$  can also be expressed as

$$r_x = F^{-1} \left[ \frac{F_1(k)F_2^*(k)}{S_1S_2} \right] \quad (5)$$

or

$$r_x = \int_{-\infty}^{\infty} \frac{F_1(k)F_2^*(k)e^{i2\pi kx}}{S_1S_2} dk, \quad (6)$$

where  $k$  is the wave number corresponding to the  $x$ -coordinate,  $F_1$  is the Fourier transform of  $f_1(x)$ ,  $F_2^*$  is the complex conjugate of the Fourier transform of  $f_2(x)$ ,  $S_1$  is the standard deviation of  $f_1$ ,  $S_2$  is the standard deviation of  $f_2$ , and  $F^{-1}$  represents the inverse Fourier transform. The displacement  $\Delta\vec{x}$  of the second waveform can be estimated by the location of the peak of the cross-correlation function plotted as a function of  $x_{lag} = -x$ . For computing efficiency, the fast Fourier transform (FFT) can be used instead of the Fourier transform to estimate the displacement  $\Delta\vec{x}$ . Let  $f_1(x)$  and  $f_2(x)$  be discrete functions, and  $N$  the number of data points in the  $x$ -direction. Then FFT of  $f_1(x)$ ,  $FFT_1$  can be expressed as

$$FFT_1 = \frac{\sum_{i=1}^N f_1 e^{-\frac{i2\pi kx}{N}}}{N}. \quad (7)$$

Similarly, FFT of  $f_2(x)$ ,  $FFT_2$  can be expressed as

$$FFT_2 = \frac{\sum_{i=1}^N f_2 e^{-\frac{i2\pi kx}{N}}}{N}. \quad (8)$$

Using  $FFT_1$  and complex conjugate of  $FFT_2$ , the cross-correlation function  $r_x$  can be expressed as

$$r_x = \frac{FFT^{-1}(FFT_1 FFT_2^*)}{S_1 S_2}, \quad (9)$$

where  $S_1$  is the standard deviation of  $f_1$ ,  $S_2$  is the standard deviation of  $f_2$ , and  $FFT^{-1}$  represents the inverse fast Fourier transform. Since lidar backscatter images are 2-dimensional (2-D), the functions  $f_1(x, y)$  and  $f_2(x, y)$  are used to express backscatter intensities. In this case, the 2-D cross-correlation function,  $r_{x,y}$  is defined. Let  $N_x$  be the number of points in the x-direction,  $N_y$  be the number of points in the y-direction,  $k_x$  be the wavenumber corresponding to the x-coordinate, and  $k_y$  be the wavenumber corresponding to the y-coordinate. Then  $r_{x,y}$  can be expressed as

$$r_{x,y} = \frac{FFT^{-1}(FFT_1 FFT_2^*)}{S_1 S_2}, \quad (10)$$

where

$$FFT_1 = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} f_1(x, y) e^{-i2\pi(\frac{k_x x}{N_x} + \frac{k_y y}{N_y})}}{N_x N_y} \quad (11)$$

and

$$FFT_2 = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} f_2(x, y) e^{-i2\pi(\frac{k_x x}{N_x} + \frac{k_y y}{N_y})}}{N_x N_y}. \quad (12)$$

The displacement  $\Delta\vec{r}$  between the two frames is estimated by identifying the location of the maximum of the cross-correlation function (the ‘‘peak’’) plotted as a function of  $x_{lag} = -x$  and  $y_{lag} = -y$ .

## Wind Velocity Fields

Figure 12 shows a possible arrangement of streamlines<sup>5</sup> of the wind velocity field in the lower atmosphere. Spatial variability of the wind velocity field is common (Venditti et al., 2013) and in this thesis a small selection of idealized, simple kinematic flows were selected: uniform flow, converging flow, diverging flow, rotational flow (vortex), and shear flow. The simplest way to represent these flow fields is to use analytical functions to create these fields. For example, if the flow is

<sup>5</sup>A line that is parallel to the direction of the flow of a fluid.



Figure 12. An example of a possible arrangement of streamlines of the wind velocity field.

perfectly uniform, the flow field can be expressed by a constant ( $v = 10 \text{ m s}^{-1}$ ) everywhere. In the case of convergent flow, the field can be expressed by an analytical function of negative divergence. However, the real atmosphere is also turbulent, so the effects of turbulence must be taken into account (Pope, 2000). To make more realistic estimations of the wind velocity fields, one can add synthetic turbulent perturbations produced by a model described by Mann (1994) and Mann (1998) to the fields produced by the analytic functions. The resulting flow fields can be used to move synthetic aerosol features and generate a pair of synthetic backscatter images. In addition, one can easily calculate the mean and the standard deviation of the velocity fields, and compare them with the velocities estimated by the cross-correlation algorithm. Thus, the synthetic wind velocity fields are useful to test the performance of the cross-correlation algorithm.

### Literature Review

The cross-correlation method has been applied in various fields of study. For example, Leese et al. (1971) applied it to geosynchronous satellite images to derive wind from cloud motion. It has also been used to find the glacier surface velocity estimation. According to Schubert et al. (2010), the glacier surface velocity can be estimated by using two image pairs separated by 11 days, acquired by using

the German sensor TerraSAR-X, and applying traditional cross-correlation optimization. The glacier surface velocity estimation was performed using the cross-correlation optimization and a dense image-matching algorithm based on complex wavelet decomposition, and the results show that the cross-correlation method is probably preferable to the wavelet-based algorithm for the glacier surface velocity estimation using repeat TerraSAR-X images (Schubert et al., 2013).

The cross-correlation technique can also be used to estimate surface displacements on mass movements from repeat optical images (Debella-Gilo and Kääh, 2011). According to Debella-Gilo and Kääh (2012), three typical mass movement types, rockglacier creep, glacier flow, and land sliding, were estimated by using the least square matching (LSM) and normalized cross-correlation, and the accuracy of displacement estimation improved by over 90% under ideal (simulated) circumstances.

The cross-correlation method can also be used to monitor seismic activities. Yun et al. (2009) applied the cross-correlation method to hydroacoustic data from the International Monitoring System arrays in the Indian Ocean to study the earthquake sequence.

Particle image velocimetry (PIV),<sup>6</sup> is another field of study that utilizes this technique. The direct cross-correlation of image fields was introduced to improve the performance of PIV in measuring instantaneous velocity fields (Keane and Adrian, 1992). One of the examples of PIV schemes is to estimate bubble velocity fields. According to Cheng et al. (2005), the recursive cross-correlation was the best measurement of bubble motion in bubble plumes among several PIV schemes tested. Another example is flow measurements in a blood-perfused

<sup>6</sup>An optical method of flow visualization to obtain instantaneous velocity and related properties in fluids.

collagen vessel. According to Antoine et al. (2013), advanced PIV cross-correlation methods were applied to measure the flow velocities in blood perfused hydrated tissue-representative microvessels using x-ray images.

In addition to these fields of study, the cross-correlation algorithm has been applied several times to atmospheric lidar data to remotely estimate wind velocity profiles in the planetary boundary layer. The earliest work was published by Eloranta et al. (1975). The radial and transverse components of the wind velocity in the convective boundary layer was estimated and compared to data from a bivane anemometer installed on a tower and within the scanning region of the lidar (Kunkel et al., 1980). The lidar measurements of wind velocity in the atmospheric boundary layer has also been compared to winds derived from radar-tracked pilot balloons (Sroga et al., 1980). Other related works include Hooper and Eloranta (1986), Kolev et al. (1988), Sasano et al. (1982) and Piironen and Eloranta (1995). Migration of the technique to derive two-component horizontal vector wind fields was done by Mayor and Eloranta (2001). Comparison of the resulting vectors to sonic anemometer data was done by Mayor et al. (2012) and with Doppler lidar data by Mayor et al. (2013). While significant experimental work has been conducted and collectively suggests viability of the technique, the performance of the technique has not been tested prior to this thesis research with simulated aerosol and wind fields.

A distinction is that in PIV discrete particles exist with relatively large areas of nothing between them. The velocity is determined based only on the movement of the particles which are good tracers of the flow. Lidar images do not show individual particles, but contain backscatter intensity everywhere. Therefore, the wind velocity is estimated based on the movement of macroscopic aerosol features (i.e., plumes). When computing the cross-correlation function, all areas

within the interrogation window contribute to the cross-correlation function but not equally. For example, the brighter aerosol features are more likely to dominate over the dimmer ones. Therefore, it is possible that the motion of the bright features biases the answer. These features have larger powers (product of the area and the intensity), compared to the surrounding regions. Thus, the hypothesis of this thesis is that the displacement vector calculated by the cross-correlation algorithm is more accurately described as the power-weighted average velocity, instead of the area-weighted average velocity described by Schols and Eloranta (1992).

### Purpose of Study

As described in the previous section, the cross-correlation algorithm has been applied many times previously to real atmospheric lidar data and efforts made to validate the results by comparing with other velocity measurements from other remote and in situ sensors. Such experiments tend to be infrequent, expensive, and present challenges when making the comparisons of wind velocities. For example, the accuracy, precision, location, and resolution of the different forms of wind measurements usually differs substantially. Furthermore, data collection requires significant periods of time to sample a variety of wind and aerosol conditions.

By creating synthetic aerosol and wind fields in a computer, input data to the cross-correlation is controlled rather than dependent on the atmosphere. In other words, the methods used herein allow the algorithm to be tested in a highly controlled fashion. Furthermore, the velocity field for the entire block area is known and a measure of the central tendency (such as the mean) of the entire velocity field can be calculated. Because of the uniqueness of horizontally scanning lidar aerosol data, no other form of wind measurement can provide the spatial wind field that is needed to calculate the correct answer. Therefore, the purpose of this study

is to evaluate the performance of a cross-correlation algorithm from a new approach involving the generation of synthetic images and wind fields.

### Hypothesis

The cross-correlation algorithm is based on finding a single peak location of the cross-correlation function, described by Eloranta et al. (1975). According to Schols and Eloranta (1992), the displacement vectors calculated by the cross-correlation algorithm is equal to the area-averaged velocities of the wind. However, the peak of the cross-correlation function is likely sensitive to the motion of predominant features in the interrogation window that have larger size and higher intensities.

## CHAPTER II

### ANALYTICAL STUDIES

In practice, the cross-correlation functions for lidar backscatter images are usually calculated numerically by the fast Fourier transform (FFT). However, for some simple cases, one can find the cross-correlation function analytically, and find an exact solution of the resultant displacement vector. In this section, the cross-correlation algorithm is limited to two types of highly idealized features, a Gaussian feature and a rectangular pulse, placed in an interrogation window. The Gaussian feature has an intensity profile that obeys the Gaussian function. Gaussian is a reasonable idealization because the turbulent diffusion of aerosol particles results in Gaussian distributions of particle concentration in the case of idealized uniform flow with homogeneous turbulence (Arya, 1999). The rectangular pulse is a feature with constant intensity. Rectangular features are of interest because of the discrete nature of digital data. Although the structure of aerosol features in actual lidar backscatter images are much more complex, these cases show how the cross-correlation function is calculated and they give some insights about the performance of the cross-correlation algorithm for extremely idealized cases.

## Analytical Approaches Using a Gaussian Feature

### Gaussian Integrals

To calculate the cross-correlation functions of such backscatter images, the following integral must be evaluated.

$$I = \int_{-\infty}^{\infty} e^{-\lambda x^2} dx. \quad (1)$$

This type of integral can be evaluated analytically. (Details of the derivation are in Appendix A.) The result is

$$I = \int_{-\infty}^{\infty} e^{-\lambda x^2} dx = \sqrt{\frac{\pi}{\lambda}}. \quad (2)$$

Suppose the Gaussian function is instead  $e^{-\lambda(x-c)^2}$  where  $c$  is a constant. Then, the position of the peak is shifted  $c$  units from the origin. If  $c$  is positive, the peak is shifted to the positive  $x$  direction. On the other hand, if  $c$  is negative, the peak is shifted to the negative  $x$  direction. However, the size and the shape of the Gaussian function is unaltered, so the result of the integral must be the same. That is

$$\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}. \quad (3)$$

This equation is also helpful to find the cross-correlation function analytically.

### One-Dimensional Case in a Uniform Flow

Suppose that there is only one Gaussian feature in a 1-D space, and the Gaussian feature is at the origin,  $x = 0$ , at time  $t_1$ . Then, the backscatter image

$f_1(x)$  at time  $t_1$  can be expressed as

$$f_1(x) = I_0 e^{-x^2}, \quad (4)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Figure 13a shows the backscatter image at time  $t_1$  in the units of  $I_0$ . In this plot, the vertical axis represents the backscatter intensity, and the horizontal axis represents the location in the 1-dimensional space. Suppose further that the Gaussian feature moves 10 units to the right in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image  $f_2(x)$  at time  $t_2$  can be expressed as

$$f_2(x) = I_0 e^{-(x-10)^2}, \quad (5)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Figure 13b shows the backscatter image at time  $t_2$ .

The 1-D cross-correlation function,  $r_x$ , applied to these images can be calculated analytically. (Details of the derivation are in Appendix A.) The result is

$$r_x = \sqrt{\frac{\pi}{2}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{2}}, \quad (6)$$

where  $I_0$  is the peak intensity of the Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x)$ , and  $S_2$  is the standard deviation of  $f_2(x)$ . Before plotting the cross-correlation function, let  $x_{lag} = -x$  for the lag. Then the cross-correlation function  $r_{x_{lag}}$  can be expressed as

$$r_{x_{lag}} = \sqrt{\frac{\pi}{2}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(-x_{lag}+10)^2}{2}}, \quad (7)$$

or

$$r_{x_{lag}} = \sqrt{\frac{\pi}{2}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x_{lag}-10)^2}{2}}. \quad (8)$$

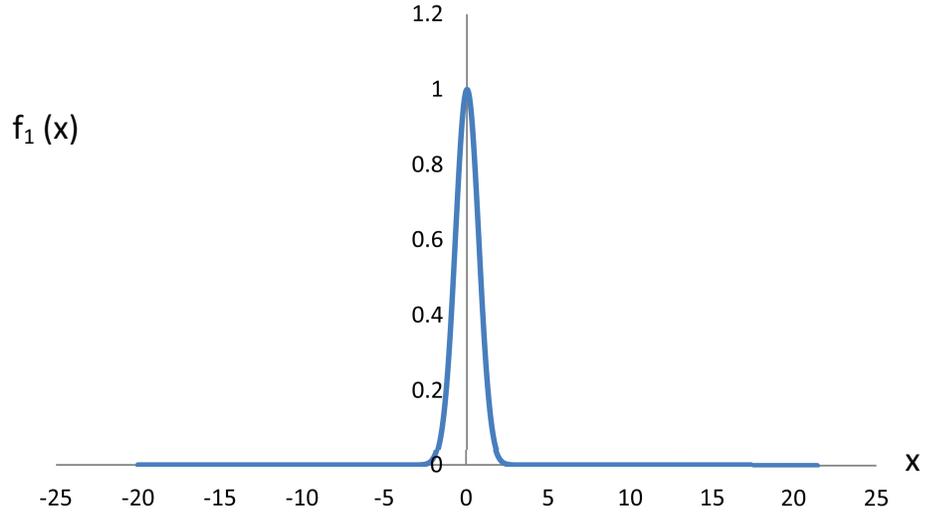
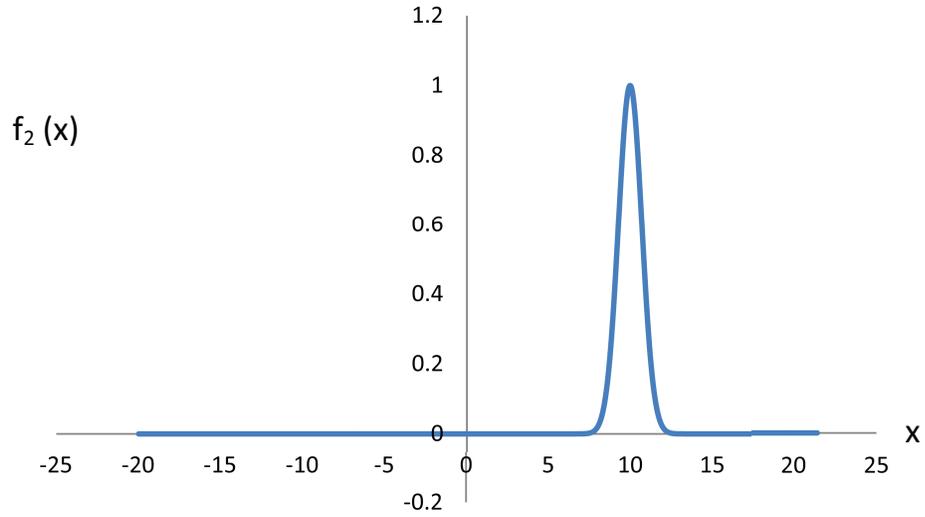
(a) 1-D Gaussian feature at  $t_1$ .(b) 1-D Gaussian feature at  $t_2$ .

Figure 13. 1-dimensional Gaussian feature moving at constant velocity. At time  $t_1$ , the center of the Gaussian feature is at the origin. The Gaussian feature is moved 10 units in the positive  $x$ -direction in the time interval  $\Delta t = t_2 - t_1$ .

That is, the cross-correlation function of 1-D Gaussian feature moving at constant rate is also a 1-D Gaussian function. However, the Gaussian is broadened by the term  $e^{-\frac{(x_{lag}-10)^2}{2}}$  rather than  $e^{-(x_{lag}-10)^2}$ . Figure 14 shows the cross-correlation function  $r_x$  as a function of  $x_{lag}$  for the 1-D Gaussian feature displaced by 10 units in the time interval  $\Delta t = t_2 - t_1$ .

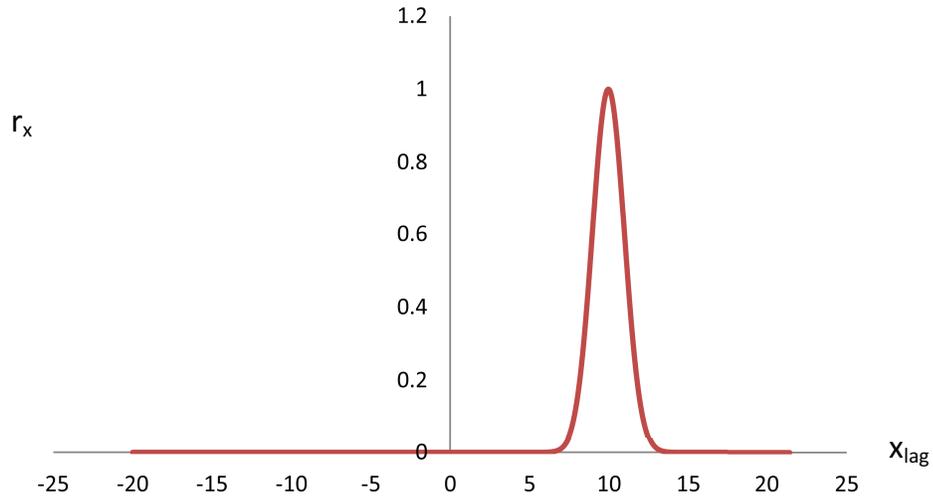


Figure 14. The cross-correlation function for given 1-D Gaussian feature. The peak location of the cross-correlation function represents the displacement, 10 units in the positive x-direction.

The peak location of the cross-correlation function  $r_{x_{lag}}$  represents the displacement vector  $\Delta\vec{x}$  in the time interval  $t_1$  and  $t_2$ . From Figure 14, the location of the peak of the cross-correlation function is shown at  $x_{lag} = 10$ . From this, one finds that the resultant displacement vector  $\Delta\vec{x}$ , calculated by the cross-correlation algorithm, is shown at 10 units in the positive x-direction. Since the resultant displacement vector is exactly equal to the displacement vector of the given

Gaussian feature, one can conclude that the cross-correlation algorithm works perfectly for a 1-D Gaussian feature moving at constant velocity.

Two-Dimensional Case in  
a Uniform Flow

The concepts of the 1-D case, described above, can be extended to the 2-D case. Suppose that there is only one Gaussian feature in a 2-D space, and the Gaussian feature is at the origin,  $(x, y) = (0, 0)$ , at time  $t_1$ . Then, the backscatter image  $f_1(x, y)$  at time  $t_1$  can be expressed as

$$f_1(x, y) = I_0 e^{-(x^2+y^2)}, \quad (9)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Figure 15a shows the backscatter image at time  $t_1$ , in the units of  $I_0$ . In this plot, the backscatter intensity is expressed by different colors. The red color represents the higher intensity compared to the blue color. Suppose further that the Gaussian feature moves 10 units to the right (positive x direction) and 10 units upward (positive y direction) in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image  $f_2(x, y)$  at time  $t_2$  can be expressed as

$$f_2(x, y) = I_0 e^{-[(x-10)^2+(y-10)^2]}, \quad (10)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Figure 15b shows the backscatter image at time  $t_2$ . The 2-D cross-correlation function  $r_{x,y}$  applied to these images can be calculated analytically, similar to that of 1-D Gaussian features (Details of the derivation are in Appendix A.) The result is

$$r_{x,y} = \frac{\pi I_0^2}{2S_1 S_2} e^{-\left[\frac{(x+10)^2+(y+10)^2}{2}\right]}, \quad (11)$$

where  $I_0$  is the peak intensity of the Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ . Next, let  $x_{lag} = -x$  and  $y_{lag} = -y$  for the lags. Then the cross-correlation function  $r_{x_{lag}, y_{lag}}$  can be expressed as

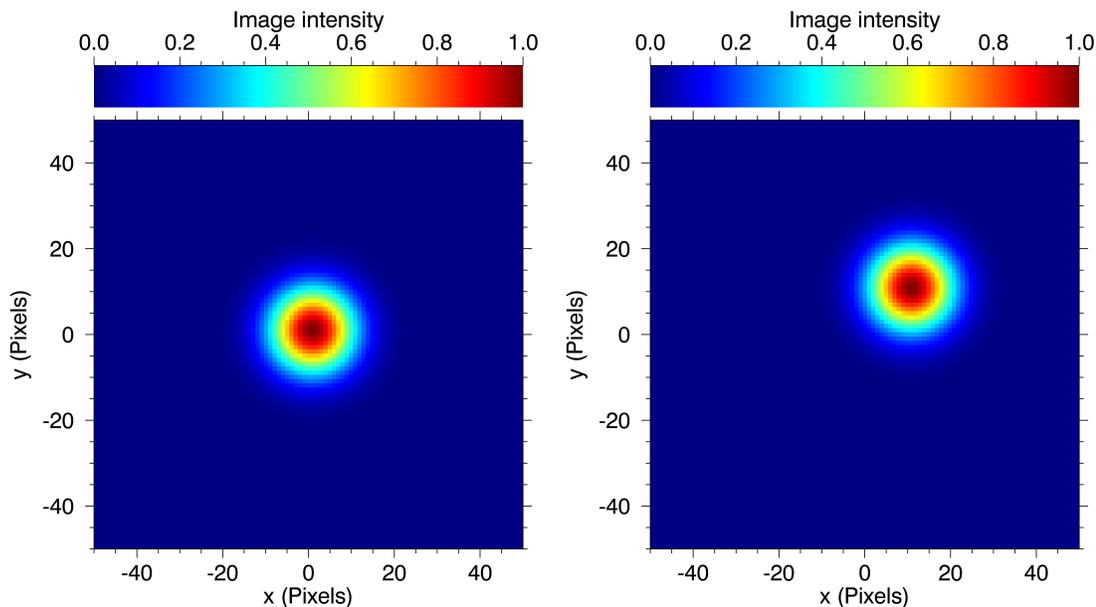
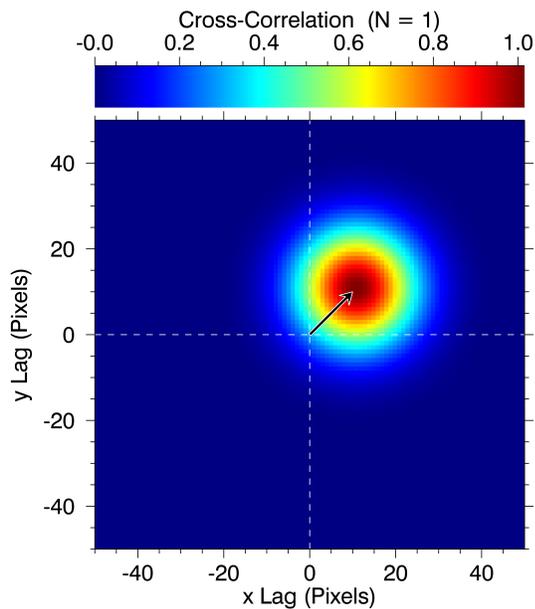
$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{2S_1 S_2} e^{-\left[\frac{(-x_{lag}+10)^2 + (-y_{lag}+10)^2}{2}\right]} \quad (12)$$

or

$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{2S_1 S_2} e^{-\left[\frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{2}\right]}. \quad (13)$$

That is, as in the 1-D case, the cross-correlation function of 2-D Gaussian feature moving at a constant rate is also a 2-D Gaussian function and the Gaussian is broadened by the term,  $e^{-\left[\frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{2}\right]}$  rather than  $e^{-[(x_{lag}-10)^2 + (y_{lag}-10)^2]}$ .

Figure 15c shows the cross-correlation function  $r_{x_{lag}, y_{lag}}$  as a function of  $x_{lag}$  and  $y_{lag}$  for the 2-D Gaussian feature displaced by 10 units to the right (positive x direction) and 10 units upward (positive y direction) between the time interval  $\Delta t = t_2 - t_1$ . Figure 15c shows that the peak of the cross-correlation function represents the displacement of the Gaussian feature in the time interval  $\Delta t = t_2 - t_1$ . Thus, one can conclude that the cross-correlation algorithm works perfectly for a 2-D Gaussian feature moving at constant velocity, although the cross-correlation function is broadened by the term  $e^{-\left[\frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{2}\right]}$ .

(a) 2-D Gaussian feature at  $t_1$ .(b) 2-D Gaussian feature at  $t_2$ .

(c) Cross-correlation function.

Figure 15. 2-D Gaussian feature moving at constant velocity. At time  $t_1$ , the center of the Gaussian feature is at the origin. The Gaussian feature is moved 10 units in the positive x-direction and 10 units in the positive y-direction in the time interval  $\Delta t = t_2 - t_1$ . The peak location of the cross-correlation function represents the displacement.

Higher Dimensional Case in  
a Uniform Flow

One can extend the discussion for a 3-D Gaussian feature at the origin,  $(x, y, z) = (0, 0, 0)$ , that is moved to the coordinates  $(x, y, z) = (10, 10, 10)$ , in the time interval  $\Delta t = t_2 - t_1$ . In this case, the backscatter images  $f_1(x, y, z)$  and  $f_2(x, y, z)$  can be expressed as

$$f_1(x, y, z) = I_0 e^{-(x^2+y^2+z^2)} \quad (14)$$

$$f_2(x, y, z) = I_0 e^{-[(x-10)^2+(y-10)^2+(z-10)^2]}, \quad (15)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. The cross-correlation function applied for these images can be calculated the same procedure with triple integrals. Let  $x_{lag} = -x$ ,  $y_{lag} = -y$ , and  $z_{lag} = -z$  for the lags. Then, the cross-correlation function  $r_{x_{lag}, y_{lag}, z_{lag}}$  can be expressed as

$$r_{x_{lag}, y_{lag}, z_{lag}} = \frac{\pi^{\frac{3}{2}} I_0^2}{2^{\frac{3}{2}} S_1 S_2} e^{-\left[ \frac{(x_{lag}-10)^2 + (y_{lag}-10)^2 + (z_{lag}-10)^2}{2} \right]}, \quad (16)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x, y, z)$ , and  $S_2$  is the standard deviation of  $f_2(x, y, z)$ . As previous cases, the cross-correlation function  $r_{x_{lag}, y_{lag}, z_{lag}}$  is also the Gaussian that is broadened by the term

$$e^{-\left[ \frac{(x_{lag}-10)^2 + (y_{lag}-10)^2 + (z_{lag}-10)^2}{2} \right]} \quad (17)$$

rather than

$$e^{-[(x_{lag}-10)^2+(y_{lag}-10)^2+(z_{lag}-10)^2]}, \quad (18)$$

and the peak location of  $r_{x,y,z}$  represents the displacement of the Gaussian feature in the time interval  $\Delta t = t_2 - t_1$ . Finally, this section is concluded by generalizing the cross-correlation function  $r_N$  for N-dimensional (N-D) Gaussian feature moving at the constant velocity. From the results,

$$r_{x_{lag}} = \sqrt{\frac{\pi}{2}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x_{lag}-10)^2}{2}} \quad (19)$$

is the equation for the 1-D case,

$$r_{x_{lag},y_{lag}} = \frac{\pi I_0^2}{2 S_1 S_2} e^{-\left[\frac{(x_{lag}-10)^2+(y_{lag}-10)^2}{2}\right]} \quad (20)$$

is for the 2-D case, and

$$r_{x_{lag},y_{lag},z_{lag}} = \frac{\pi^{\frac{3}{2}} I_0^2}{2^{\frac{3}{2}} S_1 S_2} e^{-\left[\frac{(x_{lag}-10)^2+(y_{lag}-10)^2+(z_{lag}-10)^2}{2}\right]} \quad (21)$$

is for the 3-D case. From the inspection of these expressions, the cross-correlation function  $r_N$  for N-D case can be expressed as

$$r_N = \left( \frac{\pi^{\frac{N}{2}} I_0^2}{2^{\frac{N}{2}} S_1 S_2} \right) e^{-\left[\frac{\sum_{i=1}^N (x_{i_{lag}} - d_i)^2}{2}\right]}, \quad (22)$$

where  $I_0$  is the peak intensity of the N-D Gaussian feature,  $S_1$  and  $S_2$  are the standard deviations of the images,  $x_{i_{lag}}$  are the lags for the i-th dimensions, and  $d_i$  are the i-th components of the displacement vector of the Gaussian feature. Peak locations of these functions represent the displacement of given Gaussian features. These results show that the cross-correlation algorithm works perfectly for a Gaussian feature (any dimension) moving at the constant velocity, although the cross-correlation function is broadened.

One-Dimensional Diffused Gaussian  
Feature in a Uniform Flow

Suppose a 1-D Gaussian feature is moving at a constant rate in the positive x-direction and at the same time diffused uniformly. In this case, it is also possible to find the cross-correlation function analytically. Here one can use the 1-D Gaussian feature at the origin at time  $t_1$ . Then, the backscatter image  $f_1(x)$  at time  $t_1$  can be expressed as

$$f_1(x) = I_0 e^{-x^2}, \quad (23)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Figure 16a shows the backscatter image at time  $t_1$ , in the units of  $I_0$ . Suppose further that the Gaussian feature moves 10 units to the right in the time interval  $\Delta t = t_2 - t_1$  and is diffused uniformly. Then, the backscatter image  $f_2(x)$  at time  $t_2$  can be expressed as

$$f_2(x) = \frac{I_0}{\sqrt{2}} e^{-\frac{(x-10)^2}{2}}, \quad (24)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. The factor  $\frac{1}{2}$  in the exponential term represents diffusion of the Gaussian feature and the coefficient  $\frac{1}{\sqrt{2}}$  normalizes the Gaussian feature. That is,

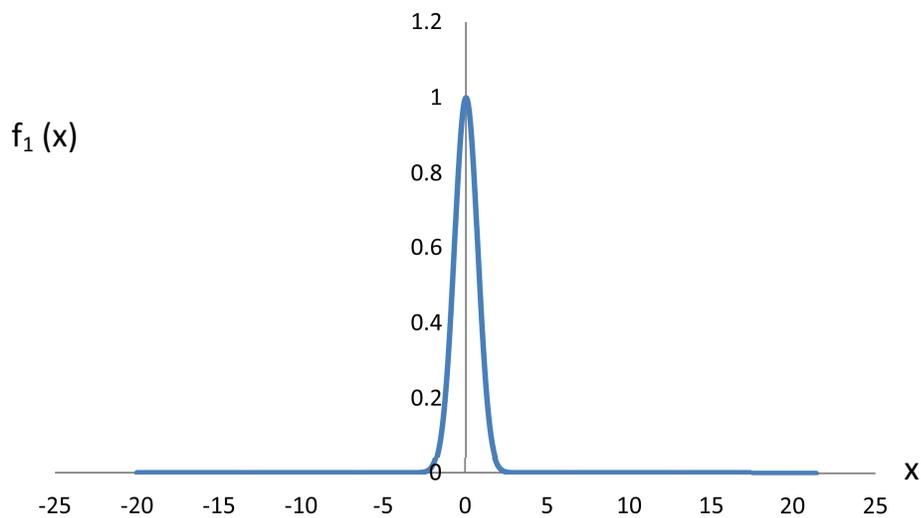
$$\int_{-\infty}^{\infty} f_2(x) dx = \int_{-\infty}^{\infty} f_1(x) dx. \quad (25)$$

Figure 16b shows the backscatter image at time  $t_2$ .

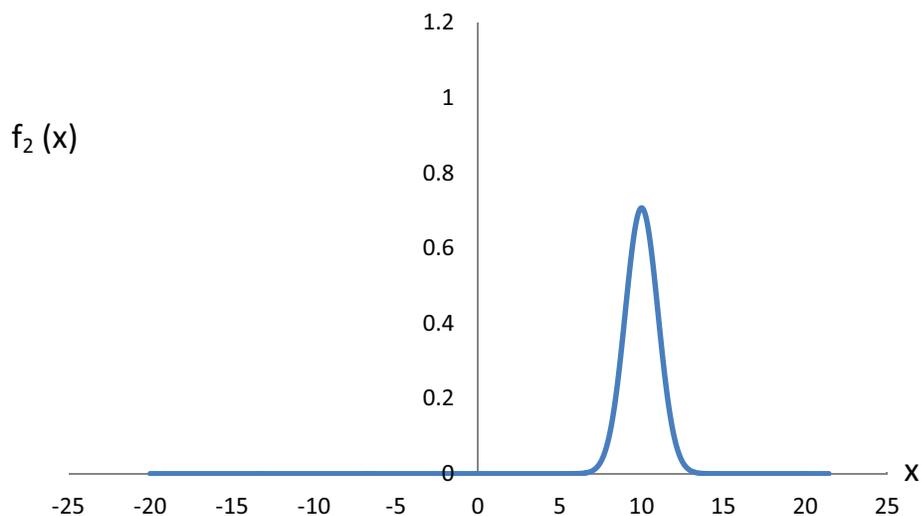
As in the previous sections, the cross-correlation function  $r_x$  applied to these images can be calculated analytically. (Details of the derivation are in

Appendix A.) The result is

$$r_x = \sqrt{\frac{\pi}{3}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{3}}. \quad (26)$$



(a) 1-D Gaussian feature at  $t_1$ .



(b) 1-D diffused Gaussian feature at  $t_2$ .

Figure 16. 1-D Gaussian feature moving at constant velocity and diffused uniformly. At time  $t_1$ , the center of the Gaussian feature is at the origin. The center Gaussian feature is moved 10 units in the positive  $x$ -direction in the time interval  $\Delta t = t_2 - t_1$ .

By defining  $x_{lag} = -x$  for the lag, the cross-correlation function  $r_{x_{lag}}$  can be expressed as

$$r_{x_{lag}} = \sqrt{\frac{\pi}{3}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(-x_{lag}+10)^2}{3}} \quad (27)$$

or

$$r_{x_{lag}} = \sqrt{\frac{\pi}{3}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x_{lag}-10)^2}{3}}, \quad (28)$$

where  $I_0$  is the peak backscatter intensity of the given Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x)$ , and  $S_2$  is the standard deviation of  $f_2(x)$ . That is, the cross-correlation function of 1-D Gaussian feature moving at constant rate and diffused uniformly is also a 1-D Gaussian function. However, the Gaussian is broadened by the term  $e^{-\frac{(x_{lag}-10)^2}{3}}$  rather than  $e^{-(x_{lag}-10)^2}$  or  $e^{-\frac{(x_{lag}-10)^2}{2}}$ . Figure 17 shows the cross-correlation function  $r_x$  as a function of  $x_{lag}$  for the 1-D Gaussian feature displaced by 10 units and diffused uniformly in the time interval  $\Delta t = t_2 - t_1$ . In Figure 17, the location of the peak of the cross-correlation function represents the displacement of the Gaussian feature in this time interval. Thus, the cross-correlation algorithm works perfectly for a 1-D Gaussian feature moving at the constant velocity and diffusing uniformly.

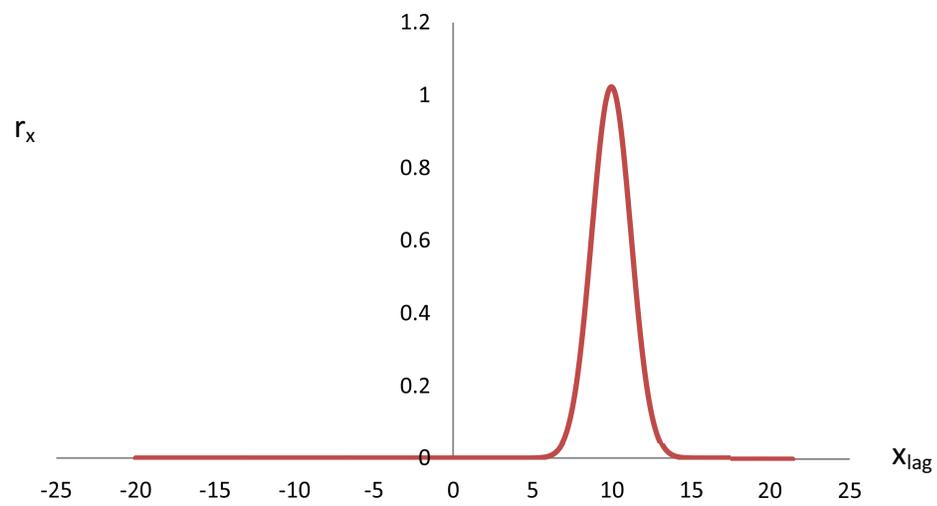


Figure 17. The cross-correlation function for given 1-D diffused Gaussian feature. The peak location of the cross-correlation function represents the displacement, 10 units in the positive x-direction.

Two-Dimensional Diffused Gaussian  
Feature in a Uniform Flow

The concepts of the 1-D diffused Gaussian feature, described above, can be extended to the 2-D case. Suppose that there is only one Gaussian feature in a 2-D space, and that the Gaussian feature is at the origin,  $(x, y) = (0, 0)$ , at time  $t_1$ . Then, the backscatter image  $f_1(x, y)$  at time  $t_1$  can be expressed as

$$f_1(x, y) = I_0 e^{-(x^2+y^2)}, \quad (29)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Figure 18a shows the backscatter image at time  $t_1$ , in the units of  $I_0$ . Suppose further that the center of the Gaussian feature moves 10 units to the right (positive x direction), 10 units to the upward (positive y direction), and the Gaussian feature diffuses uniformly in the xy-plane in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image  $f_2(x, y)$  at time  $t_2$  can be expressed as

$$f_2(x, y) = \frac{I_0}{2} e^{-\frac{[(x-10)^2+(y-10)^2]}{2}}, \quad (30)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Figure 18b shows the backscatter image at time  $t_2$ . As in the 1-D case, the factor  $\frac{1}{2}$  in the exponential term represents diffusion and the coefficient  $\frac{1}{2}$  normalizes the Gaussian feature. The 2-D cross-correlation function,  $r_{x,y}$ , applied to these images can be calculated similarly in the 1-D case. (Details of the derivation are in Appendix A.)

The result is

$$r_{x,y} = \frac{\pi I_0^2}{3S_1 S_2} e^{-\left[\frac{(x+10)^2+(y+10)^2}{3}\right]} \quad (31)$$

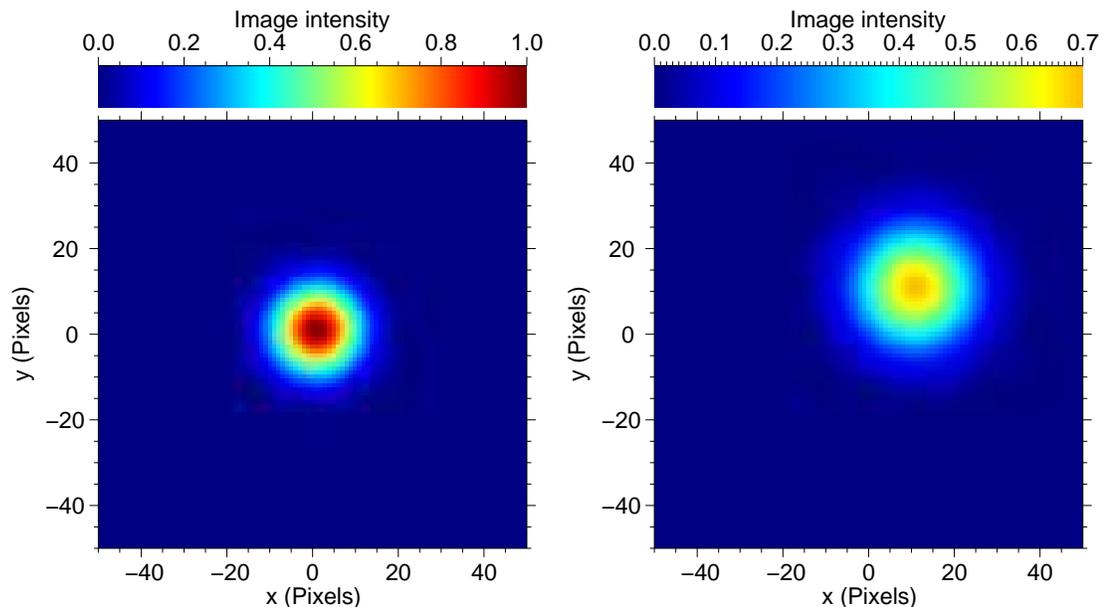
By defining  $x_{lag} = -x$  and  $y_{lag} = -y$  for the lags, the cross-correlation function  $r_{x_{lag}, y_{lag}}$  can be expressed as

$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{3S_1 S_2} e^{-\left[ \frac{(-x_{lag}+10)^2 + (-y_{lag}+10)^2}{3} \right]} \quad (32)$$

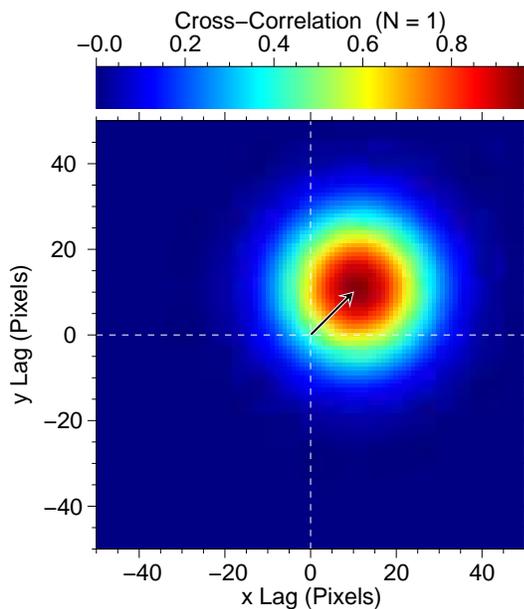
or

$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{3S_1 S_2} e^{-\left[ \frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{3} \right]}, \quad (33)$$

where  $I_0$  is the peak backscatter intensity of the given Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ . That is, as in the 1-D case, the cross-correlation function of a 2-D Gaussian feature moving at constant rate and diffused uniformly in the xy-plane is also a 2-D Gaussian function. However, the Gaussian is broadened by the term  $e^{-\left[ \frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{3} \right]}$  rather than  $e^{-[(x_{lag}-10)^2 + (y_{lag}-10)^2]}$  or  $e^{-\left[ \frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{2} \right]}$ . Figure 18c shows the cross-correlation function  $r_{x,y}$  as a function of  $x_{lag}$  and  $y_{lag}$  for the 2-D Gaussian feature displaced by 10 units to the right (positive x direction), 10 units upward (positive y direction), and diffused uniformly in the xy-plane, in the time interval  $\Delta t = t_2 - t_1$ . In Figure 18c, the peak of the cross-correlation function represents the displacement of the Gaussian feature in the time interval  $\Delta t = t_2 - t_1$ . Thus, as in the 1-D case, the cross-correlation algorithm works perfectly for a 2-D Gaussian feature moving at the constant velocity and diffusing uniformly.



(a) 2-D Gaussian feature.

(b) Diffused Gaussian feature at  $t_2$ .

(c) Cross-correlation function.

Figure 18. 2-D Gaussian feature moving at constant velocity and diffused uniformly in the  $xy$ -plane. At time  $t_1$ , the center of the Gaussian feature is at the origin. The Gaussian feature is moved 10 units in the positive  $x$ -direction and 10 units in the positive  $y$ -direction in the time interval  $\Delta t = t_2 - t_1$ . The peak location of the cross-correlation function represents the displacement.

Higher Dimensional Diffused Gaussian  
Feature in a Uniform Flow

The concepts of the 1-D diffused Gaussian feature, described above, can be extended to the N-D Gaussian feature moving at constant rate and diffused uniformly. Let  $f_1$  represent the N-D Gaussian feature at the origin

$$f_1 = I_0 e^{-(\sum_{i=1}^N x_i^2)} \quad (34)$$

and  $f_2$  represent the N-D Gaussian feature moving  $d_i$  units in each dimension and diffused uniformly.

$$f_2 = \frac{I_0}{2^{\frac{N}{2}}} e^{-\left[\frac{\sum_{i=1}^N (x_i - d_i)^2}{2}\right]} \quad (35)$$

From inspections of previous results, the cross-correlation function  $r_N$  applied to  $f_1$  and  $f_2$  can be expressed as

$$r_N = \left( \frac{\pi^{\frac{N}{2}} I_0^2}{3^{\frac{N}{2}} S_1 S_2} \right) e^{-\left[\frac{\sum_{i=1}^N (x_i - d_i)^2}{3}\right]}, \quad (36)$$

where  $I_0$  is the peak backscatter intensity of the given N-D Gaussian feature,  $S_1$  is the standard deviation of  $f_1$ ,  $S_2$  is the standard deviation of  $f_2$ , and  $d_i$  is the  $i$ -th components of the displacement of the Gaussian feature. That is, one can find the cross-correlation function and the displacement vector analytically for the N-D Gaussian feature as long as it is moving at a constant velocity and diffused uniformly in all directions. The peak of the cross-correlation function represents the displacement vector of a given Gaussian feature, and hence, the cross-correlation algorithm works perfectly in these situations.

## Analytical Approaches Using a Rectangular Pulse

### Contour Integration

As done with previous examples of Gaussian features, one can evaluate the cross-correlation function analytically for a rectangular pulse moving at constant velocity. To calculate the cross-correlation function for a rectangular pulse, one evaluates the following integral.

$$I = \int_0^{\infty} \frac{\sin(\lambda x)}{x} dx, \quad (37)$$

where  $\lambda$  is a positive constant. To evaluate this integral, the “Calculus of Residues”, described by Arfken and Weber (2001) and Mathews and Walker (1970), is applied. (Details of the derivation are in Appendix B.) The result is

$$I = \int_0^{\infty} \frac{\sin(\lambda x)}{x} dx = \frac{\pi}{2}. \quad (38)$$

This equation is helpful to evaluate the cross-correlation function for rectangular pulses moving at constant velocity.

### One-Dimensional Rectangular Pulse in a Uniform Flow

Suppose that there is only 1 rectangular pulse in a 1-D space, and the center of the rectangular pulse is at the origin,  $x = 0$ , at time  $t_1$ . Then, the backscatter image  $f_1(x)$  at time  $t_1$  can be expressed as

$$f_1(x) = \begin{cases} I_0 & : -\frac{a}{2} \leq x \leq \frac{a}{2} \\ 0 & : otherwise \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the rectangular pulse. Figure 19a shows the backscatter image at time  $t_1$ . In this plot, the vertical axis represents the backscatter intensity, and the horizontal axis represents the location in the 1-D space. Suppose further that the rectangular pulse moves 10 units to the right between the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image  $f_2(x)$  at time  $t_2$  can be expressed as

$$f_2(x) = \begin{cases} I_0 & : -\frac{a}{2} + 10 \leq x \leq \frac{a}{2} + 10 \\ 0 & : otherwise \end{cases}$$

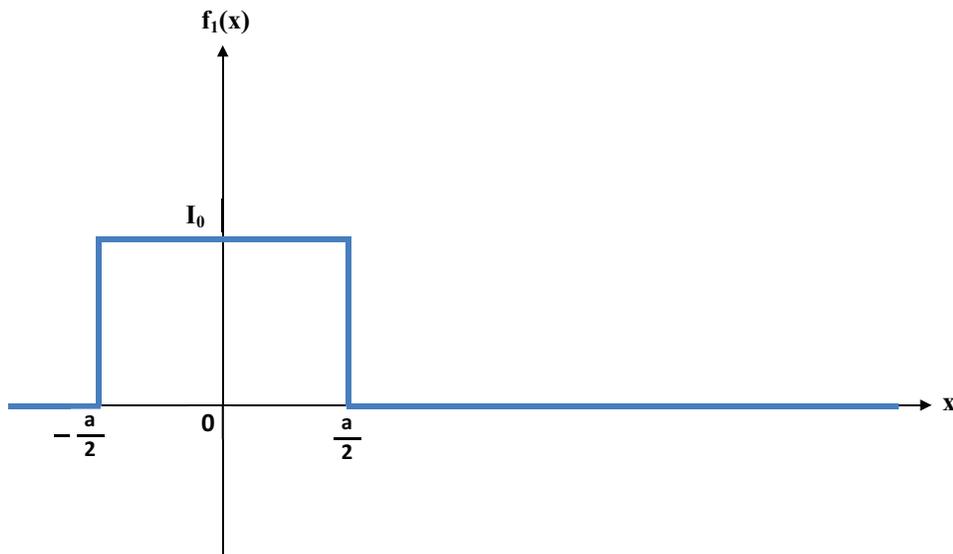
where  $I_0$  is the peak backscatter intensity of the rectangular pulse. Figure 19b shows the backscatter image at time  $t_2$ .

The 1-D cross-correlation function,  $r_{x_{lag}}$ , applied to these images can be calculated analytically. (Details of the derivation are in Appendix B.) Let  $x_{lag} = -x$ . Then the cross-correlation function  $r_{x_{lag}}$  can be expressed as

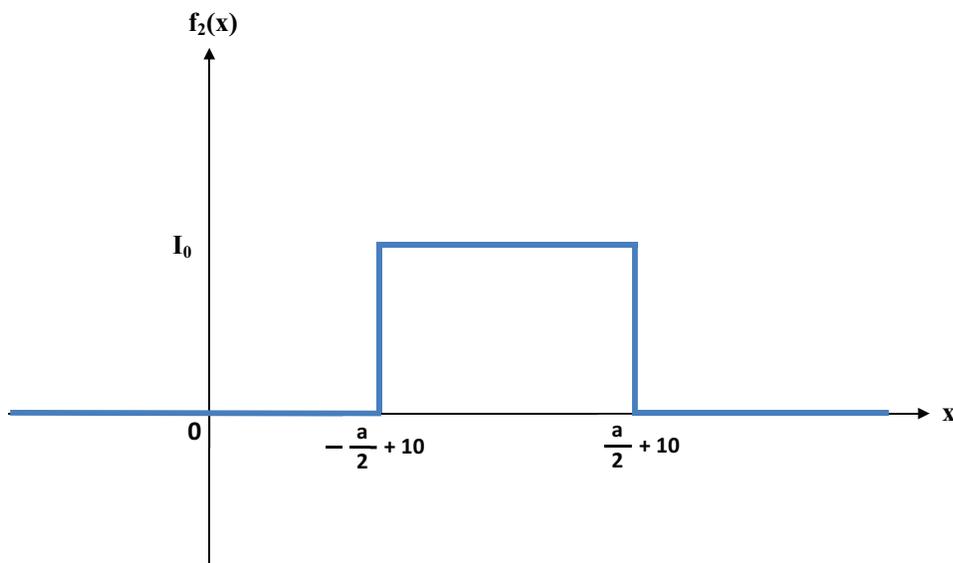
$$r_{x_{lag}} = \begin{cases} \left( \frac{I_0^2}{S_1 S_2} \right) a + \left( \frac{I_0^2}{S_1 S_2} \right) (x_{lag} - 10) & : 10 - a \leq x_{lag} \leq 10 \\ \left( \frac{I_0^2}{S_1 S_2} \right) a - \left( \frac{I_0^2}{S_1 S_2} \right) (x_{lag} - 10) & : 10 \leq x_{lag} \leq 10 + a \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the given rectangular pulse,  $a$  is the width of the rectangular pulse,  $S_1$  is the standard deviation of  $f_1(x)$ , and  $S_2$  is the standard deviation of  $f_2(x)$ . That is, the cross-correlation function of a rectangular pulse moving at a constant rate is a triangular function. The base of the triangle is  $2a$ , that is as twice as large as the width of the given rectangular pulse. The peak of the triangle is proportional to the square of the peak intensity  $I_0$  of the given rectangular pulse. Figure 20 shows the cross-correlation function  $r_x$  as a function of  $x_{lag}$  for the rectangular pulse of width  $a$  displaced by 10 units between the time

interval  $\Delta t = t_2 - t_1$ . From Figure 20, one can see that the location of the peak of the cross-correlation function represents the displacement of the rectangular pulse



(a) 1-D rectangular pulse at  $t_1$ .



(b) 1-D rectangular pulse at  $t_2$ .

Figure 19. 1-D rectangular pulse moving at constant velocity. At time  $t_1$ , the center of the rectangular pulse is at  $x = 0$ . The rectangular pulse is moved 10 units in the positive  $x$ -direction in the time interval  $\Delta t = t_2 - t_1$ .

in the time interval between  $t_1$  and  $t_2$ . Thus, the cross-correlation algorithm works perfectly for the 1-D rectangular pulse moving with constant velocity.

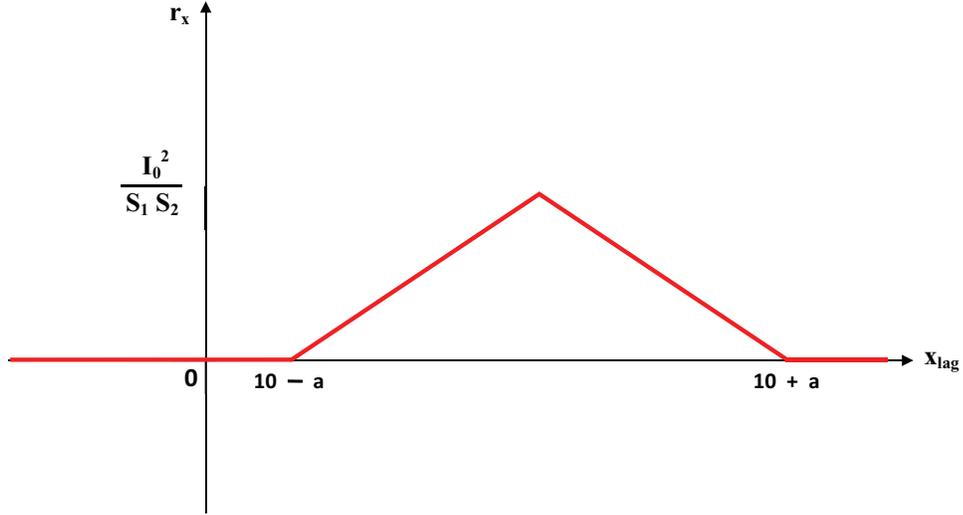


Figure 20. The cross-correlation function for a given 1-D rectangular pulse. The peak location of the cross-correlation function represents the displacement, 10 units in the positive x-direction.

### Two-Dimensional Rectangular Pulse in a Uniform Flow

The concepts of the 1-D case, described above, can be extended to the 2-D case. Suppose that there is only 1 rectangular pulse in a 2-D space, and the center of the rectangular pulse is at the origin,  $(x, y) = (0, 0)$ , at time  $t_1$ . Then, the backscatter image  $f_1(x, y)$  at time  $t_1$  can be expressed as

$$f_1(x, y) = \begin{cases} I_0 & : -\frac{a}{2} \leq x \leq \frac{a}{2} : -\frac{b}{2} \leq y \leq \frac{b}{2} \\ 0 & : otherwise \end{cases}$$

where  $I_0$  is the peak backscatter image of the rectangular pulse. Figure 21a shows the backscatter image at time  $t_1$ , in the units of  $I_0$ . In this plot, the red-color rectangle represents the rectangular pulse of the intensity  $I_0$  and the blue region represents the zero intensity. Suppose further that the rectangular pulse moves 10 units to the right (positive x direction) and 10 units upward (positive y direction) in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image  $f_2(x, y)$  at time  $t_2$  can be expressed as

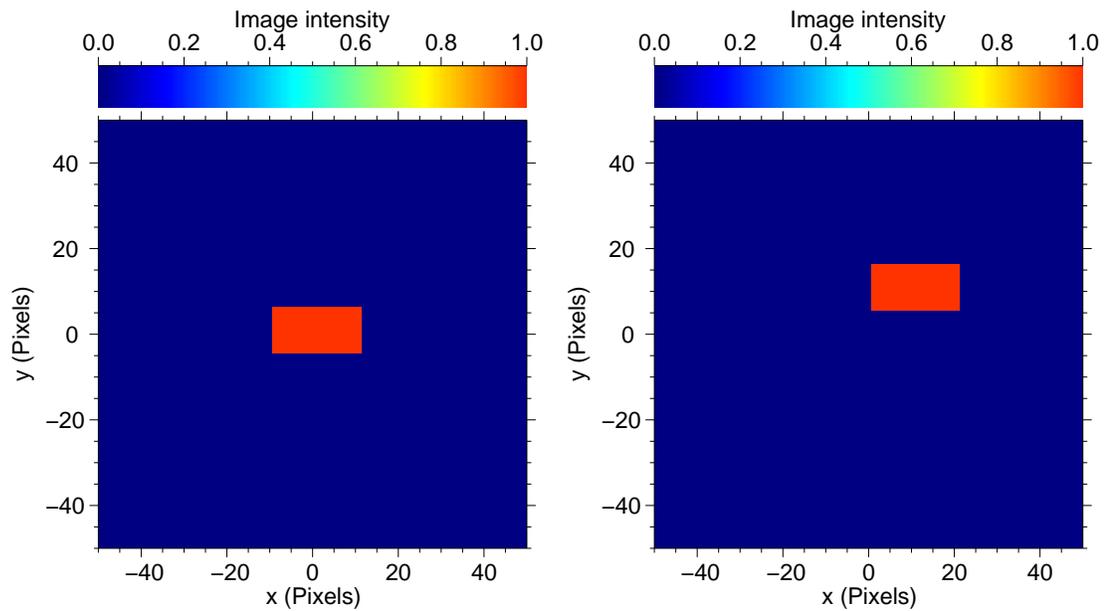
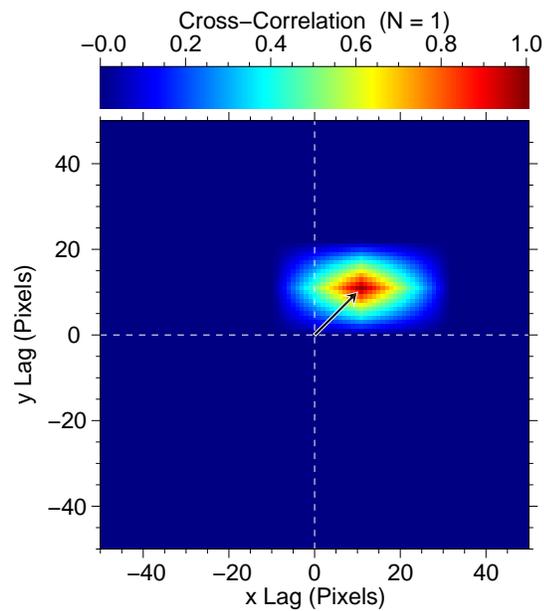
$$f_2(x, y) = \begin{cases} I_0 & : 10 - \frac{a}{2} \leq x \leq 10 + \frac{a}{2} : 10 - \frac{b}{2} \leq y \leq 10 + \frac{b}{2} \\ 0 & : otherwise \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the rectangular pulse. Figure 21b shows the backscatter image at time  $t_2$ . The 2-D cross-correlation function  $r_{x_{lag}, y_{lag}}$  applied to these images can be calculated analytically. (Details of the derivation is in Appendix B.) Let  $x_{lag} = -x$ , and  $y_{lag} = -y$ . Then the cross-correlation function  $r_{x_{lag}, y_{lag}}$  can be expressed as

$$r_{x_{lag}, y_{lag}} = \begin{cases} \left( \frac{I_0^2}{S_1 S_2} \right) [a + (x_{lag} - 10)][b + (y_{lag} - 10)] & : 10 - a \leq x_{lag} \leq 10 : 10 - b \leq y_{lag} \leq 10 \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a - (x_{lag} - 10)][b - (y_{lag} - 10)] & : 10 \leq x_{lag} \leq 10 + a : 10 \leq y_{lag} \leq 10 + b \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a + (x_{lag} - 10)][b - (y_{lag} - 10)] & : 10 - a \leq x_{lag} \leq 10 : 10 \leq y_{lag} \leq 10 + b \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a - (x_{lag} - 10)][b + (y_{lag} - 10)] & : 10 \leq x_{lag} \leq 10 + a : 10 - b \leq y_{lag} \leq 10 \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the given rectangular pulse,  $a$  is the width of the rectangular pulse,  $b$  is the height of the rectangular pulse,  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ . That is, the cross-correlation function of a rectangular pulse moving at a constant rate is a

pyramid-shape function of rhombus base. The diagonals of the rhombus base are  $2a$  and  $2b$ , twice as large as the dimensions of the given rectangular pulse. The peak of the pyramid-shape function is proportional to the square of the peak intensity  $I_0$  of the given rectangular pulse. Figure 21c shows the cross-correlation function  $r_{x,y}$  as a function of  $x_{lag}$  and  $y_{lag}$  for the rectangular pulse of width  $a = 20$  units and height  $b = 10$  units displaced by 10 units in both x and y directions in the time interval  $\Delta t = t_2 - t_1$ . From Figure 21c, one can see that the location of the peak of the cross-correlation function represents the displacement of the rectangular pulse between the time interval  $t_1$  and  $t_2$ . In addition, the diagonals of the rhombus base are  $2a = 40$  units and  $2b = 20$  units. Thus, as in the case of the 1-D rectangular pulse, the exact displacement of the 2-D rectangular pulse moving with constant velocity is correctly identified. That is, the cross-correlation algorithm works perfectly for a 2-D rectangular pulse moving with constant velocity in the xy-plane.

(a) 2-D rectangular pulse at  $t_1$ .(b) 2-D rectangular pulse at  $t_2$ .

(c) Cross-correlation function.

Figure 21. 2-D rectangular pulse moving at constant velocity. At time  $t_1$ , the center of the rectangular pulse is at the origin. The rectangular pulse is moved 10 units in the positive x-direction and 10 units in the positive y-direction in the time interval  $\Delta t = t_2 - t_1$ . The peak location of the cross-correlation function represents the displacement.

## CHAPTER III

### METHODOLOGY

To test the performance of the cross-correlation algorithm as applied to atmospheric lidar data, a 2-D synthetic wind velocity field is applied to a synthetic backscatter image to displace each pixel of the image to a new location. Next, bicubic interpolation is applied to determine the synthetic backscatter intensity at each element of the Cartesian grid. This results in a pair of synthetic backscatter images and one corresponding wind velocity field. Then, the cross-correlation algorithm is applied to this pair of images, and the displacement vector is calculated. The tests are repeated on 100 different pairs of synthetic backscatter images, and the mean and the standard deviation of the resulting displacement vectors due to different synthetic backscatter images are calculated. Finally, the mean resultant displacement vector is compared to the known mean wind velocity field. The test is performed to investigate (1) the performance of the cross-correlation algorithm for various velocity fields that are the sum of various analytical functions and turbulent fluctuations, (2) the performance of the cross-correlation algorithm for a variety of turbulent intensities, and (3) the performance of the cross-correlation algorithm in the case where a predominant feature is cut by the edge of a lidar scan. All computer programs to conduct these experiments were written in Interactive Data Language (IDL).

## Different Types of Wind Velocity Fields

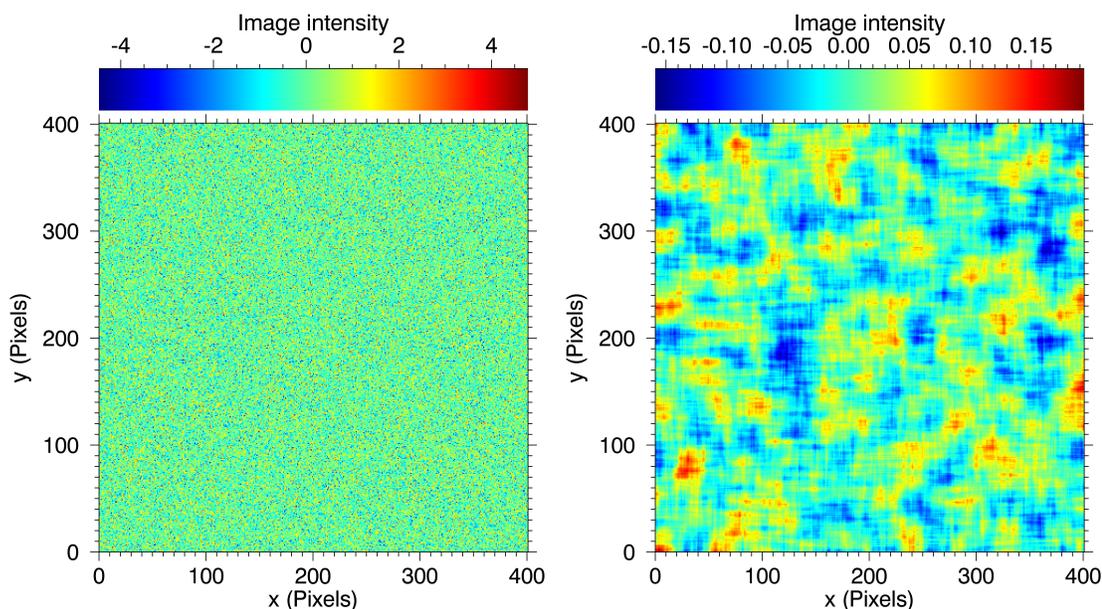
### Grid Space and Time Interval

The grid spacing  $(\Delta x, \Delta y)$  and the time interval  $\Delta t$  of the synthetic images are set so that the experiment is comparable with actual lidar backscatter images and wind velocity fields. The grid space and the time interval are adjustable parameters. For example, Mayor et al. (2012) used the grid space  $\Delta x = \Delta y = 10$  m, and the time interval  $\Delta t = t_2 - t_1$  of 17 s. Here, the grid spacing is set so that 1 pixel of an image corresponds to 10 m, and the time interval  $\Delta t = t_2 - t_1$  of 10 s. In this case, the motion of a feature of 1 pixel between two consecutive frames is equivalent to the displacement of 10 m in the time interval of 10 s, and the corresponding velocity is  $1.0 \text{ m s}^{-1}$ . Using these dimensions, it is easy to compare synthetic backscatter images and actual lidar images.

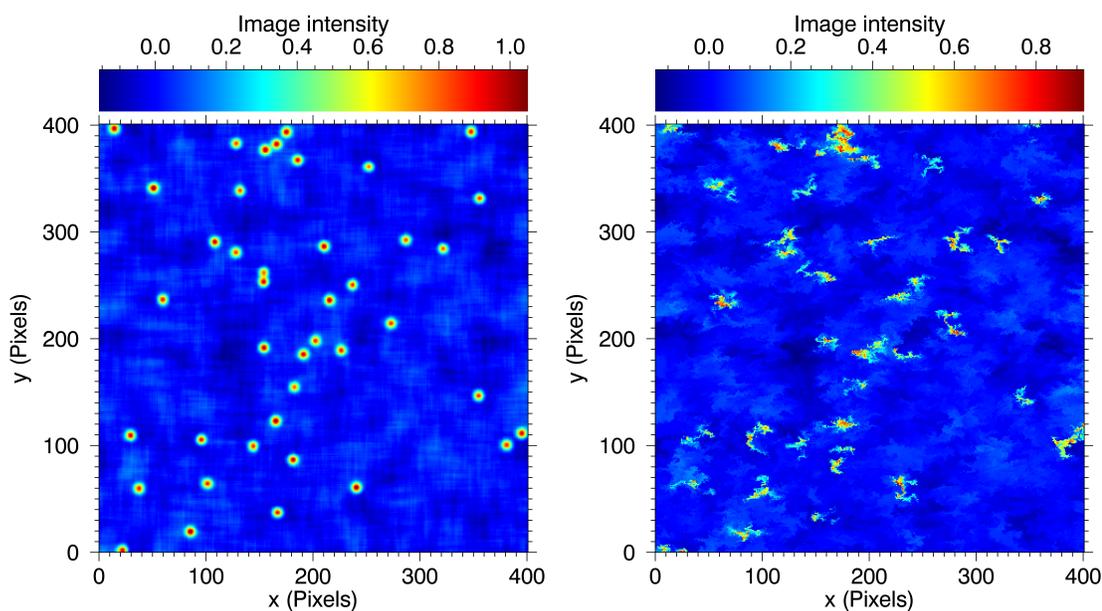
### Synthetic Backscatter Image

A synthetic backscatter image is created by generating a 2-dimensional array filled with a uniform distribution of random numbers (Figure 22a). The backscatter intensities in the image are represented by colors of the plot. The red and blue colors represent higher and lower backscatter intensities respectively. No coherent features are apparent in the image. Spatially coherent features are created by applying a  $25 \times 25$  pixel boxcar smooth shown in Figure 22b. These features mimic the background of the atmospheric aerosol. 25 pixels at 10 m per pixel produces coherent features with characteristic length scales of 250 m. Next, Gaussian features were randomly placed in the interrogation window, as shown in Figure 22c. These Gaussian features represent plumes of atmospheric aerosol from nearby sources, for example agricultural equipment working beneath the scan plane, and are a common feature in real lidar data. In real lidar backscatter

images, such predominant features are more than 10 times brighter than the background, so the intensities of these Gaussian features are about the same relative brightness as the ones found in the real lidar backscatter images. At this point, the intensities of the synthetic backscatter image are comparable to those of the real lidar data. However, the shapes of the features in the synthetic backscatter image at this point are not similar to those of the real ones. To make the synthetic backscatter image more realistic, a synthetic turbulent perturbation fields, as generated by the spectral tensor turbulent model of Mann (1994) and Mann (1998), is used to diffuse the features in the interrogation window. After the diffusion, the synthetic backscatter image becomes the one shown in Figure 22d. These images are  $400 \times 400$  pixels that correspond to  $4 \times 4$  km area. From these images  $100 \times 100$  pixels subsets are selected for input of the cross-correlation algorithm.



(a) 2-D array filled with random numbers. (b) Image after  $25 \times 25$  boxcar smoothing.



(c) Image after adding Gaussian features. (d) Image after turbulent diffusion.

Figure 22. Results of steps taken to create a synthetic backscatter image. Panel a shows the image created by 2-D array filled with random numbers. Panel b shows the image after smoothing the 2-D array with a 2-D moving average. Panel c shows the image after adding Gaussian features. Panel d is the image after applying the synthetic turbulent perturbation field. Panel d is the synthetic backscatter image for testing the performance of the cross-correlation algorithm.

### Synthetic Wind Velocity Fields

A synthetic 2-D, 2-component, wind velocity field is generated with an analytical function, or a combination of analytical function and synthetic turbulent perturbations as produced by a model described by Mann (1994) and Mann (1998). To describe such wind velocity fields, let the 2-dimensional position vector  $\vec{r} = (x, y)$ , where  $x$  and  $y$  are the east-west and the north-south components of the position vector  $\vec{r}$ . Next, let  $u(x, y)$  and  $v(x, y)$  be the east-west and the north-south components of the wind velocity  $(u, v)$ , respectively. Then,  $u(x, y)$  and  $v(x, y)$  can be expressed as

$$u(x, y) = u_0(x, y) + u'(x, y) \quad (1)$$

and

$$v(x, y) = v_0(x, y) + v'(x, y), \quad (2)$$

where  $u_0(x, y)$  and  $v_0(x, y)$  are analytical functions of  $x$  and  $y$  and  $u'(x, y)$  and  $v'(x, y)$  are turbulent perturbations. Figure 23 shows one example of wind velocity fields. Here, the wind velocity field is created by a combination of a uniform flow field (10 pixels/frames) and turbulent perturbation field. The arrows and colors in Figure 23a represent directions of the wind velocity field and speeds (magnitude of the wind velocity) at the given pixel locations, respectively. Turbulent perturbations  $u'(x, y)$  and  $v'(x, y)$  can be increased by increasing the aerodynamic roughness length,<sup>1</sup>  $z_0$ . For experiments of different velocity fields, one can use the typical value  $z_0 = 0.5$  m at the altitude of 30 m.

<sup>1</sup>A parameter used to model the vertical profile of mean horizontal wind speed near the ground. It is equivalent to the height where the wind speed theoretically becomes zero, and no longer follows a log wind profile (Stull, 1988).

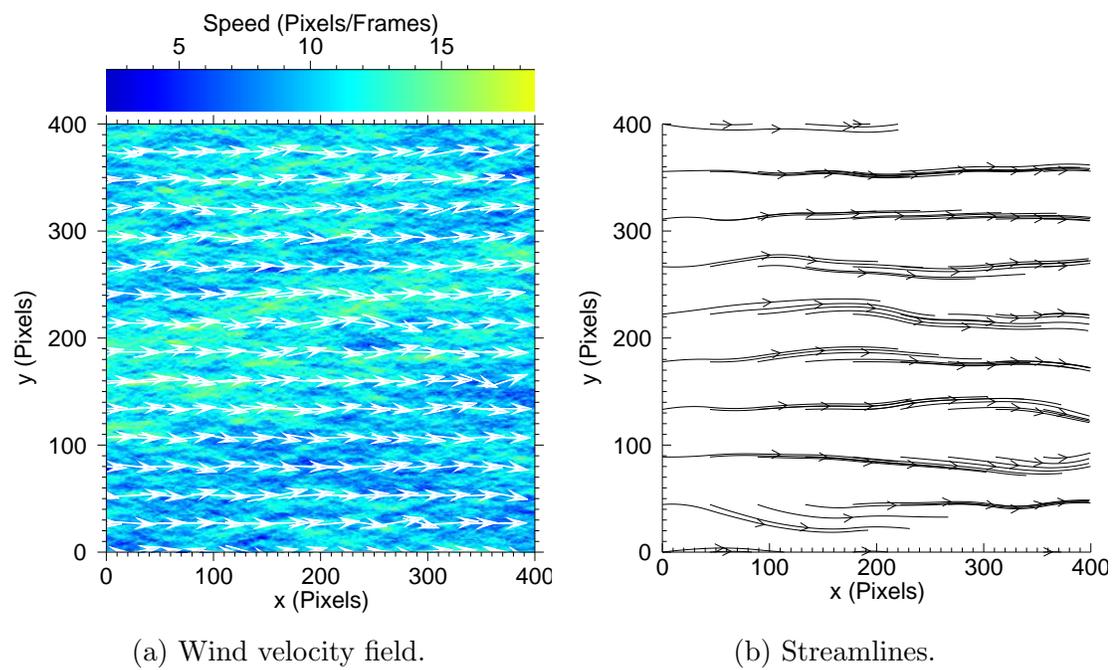
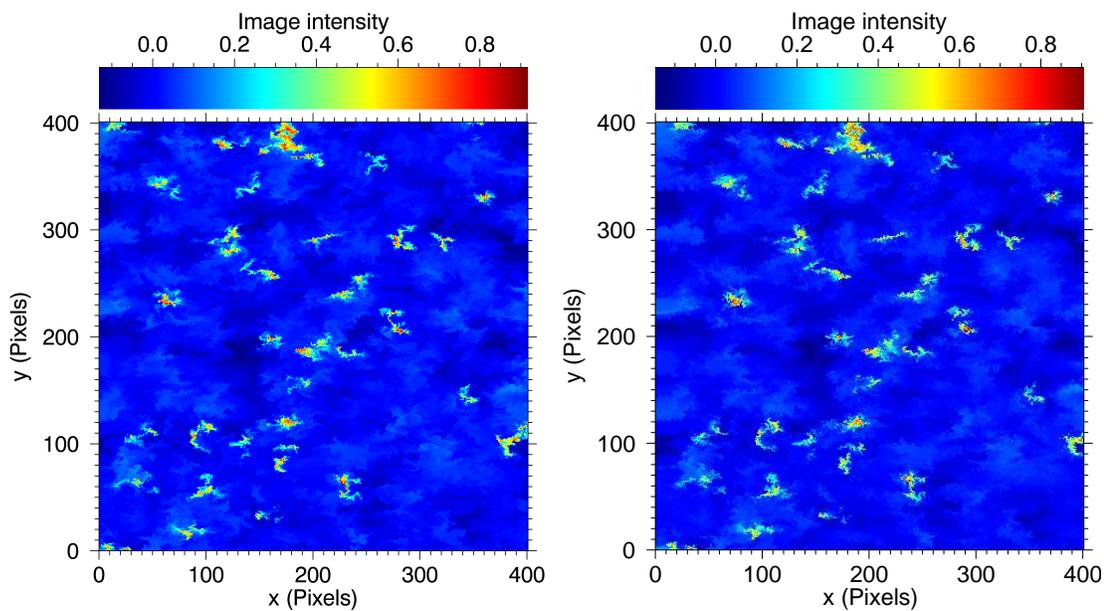


Figure 23. The velocity field created by a uniform flow field and synthetic turbulent perturbation field. Panel a and b show the velocity field and the streamlines, respectively.

### Creation of the Second Synthetic Backscatter Image

To create the second image, each pixel in the synthetic backscatter image is moved to a new location according to the wind vector at the pixel location. Since the displaced location of the pixel is not likely to fall exactly on a vertex of the Cartesian grid, bicubic interpolation is applied to create a distribution of backscatter intensities on the Cartesian grid. After applying the wind velocity field and the bicubic interpolation, the synthetic backscatter image becomes the one shown in Figure 24b. Comparing Figures 24a and 24b, one can see that the features in the first block are displaced to the right (positive x-direction) with turbulent perturbations. Some features are distorted, but one can recognize most of them from the first image.



(a) Synthetic backscatter image at  $t_1$ .

(b) Synthetic backscatter image at  $t_2$ .

Figure 24. Panel a shows the synthetic backscatter image at  $t_1$ . Panel b shows the synthetic backscatter image at  $t_2$  created by moving each pixel in Panel a to a new location according to the velocity field shown in Figure 23, and applying bicubic interpolation.

### Zero-Padding

The cross-correlation function of two images, shown in Figures 24a and 24b, can be computed using the definition of the cross-correlation function  $r_{x,y}$

$$r_{x,y} = \frac{COV_{1,2}}{S_1 S_2}, \quad (3)$$

where  $COV_{1,2}$  is the covariance of overlapped portions of  $f_1(x, y)$  and  $f_2(x, y)$  (backscatter intensities of the two images),  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ . In practice, this is done most efficiently by use of the fast Fourier transform (FFT). The ability to do this is provided by the Wiener-Khinchin theorem (Bendat and Piersol, 2011). Let  $N_x$  stand for the number of points in the x-direction,  $N_y$  the number of points in the y-direction,  $k_x$  the wavenumber corresponding to the x-coordinate, and  $k_y$  the wavenumber corresponding to the y-coordinate. Then  $r_{x,y}$  can be expressed as

$$r_{x,y} = \frac{FFT^{-1}(FFT_1 FFT_2^*)}{S_1 S_2}, \quad (4)$$

where

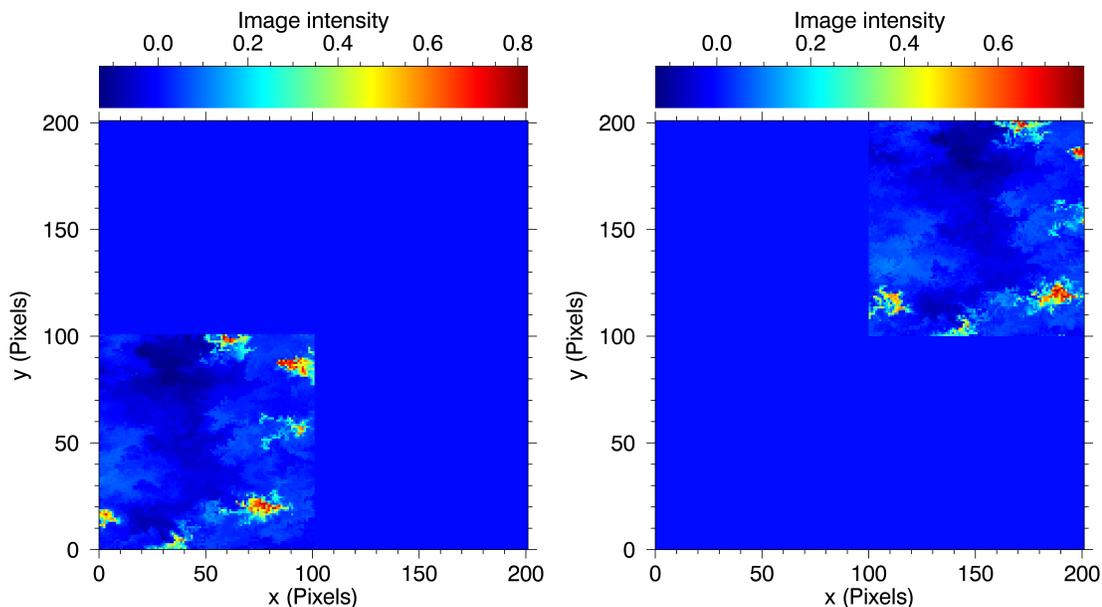
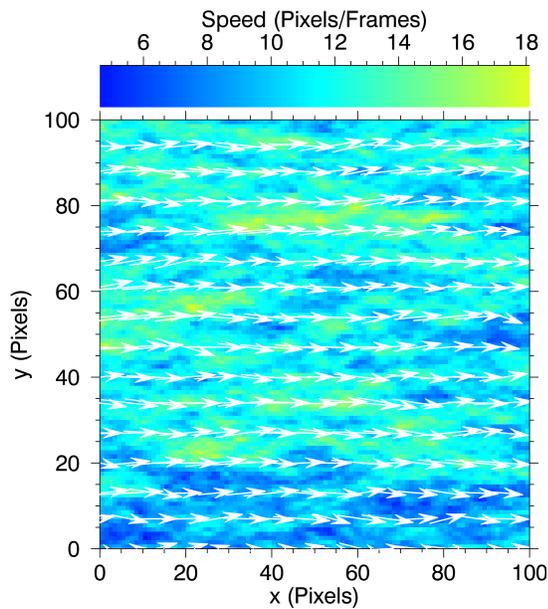
$$FFT_1 = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} f_1(x, y) e^{-i2\pi(\frac{k_x x}{N_x} + \frac{k_y y}{N_y})}}{N_x N_y} \quad (5)$$

and

$$FFT_2 = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} f_2(x, y) e^{-i2\pi(\frac{k_x x}{N_x} + \frac{k_y y}{N_y})}}{N_x N_y}. \quad (6)$$

This method is computationally more efficient than using the covariance method. However, the FFT method assumes that given images are periodic in order to integrate over finite domains (Raffel, 2007). This can be circumvented by zero-padding, that is, the interrogation images are padded with zeros over a domain

that is at least twice the size of the original signal (Bastiaans, 2000). In the case of 2-D images, the interrogation windows are padded by zeros for three quarters of its total size (Adrian and Westerweel, 2011). Using such a zero-padded image, one can avoid errors associated with the assumption of periodicity. Figure 25a represents the synthetic backscatter image before applying the wind velocity field, called “block 1”. Here, a subset of  $100 \times 100$  pixels of the synthetic backscatter image are placed at the lower left corner of the input array ( $200 \times 200$  pixels), and zeros set for all other regions. Figure 25b shows the synthetic backscatter image after applying the wind velocity field, and it is called “block 2”. As block 1,  $100 \times 100$  pixels of synthetic backscatter image is placed and set all zeros for the rest of region. However, the synthetic backscatter image is placed in the upper right of the input array, instead of the lower left. For comparison, one extracts the corresponding part of the velocity field as shown in Figure 25c, and the mean and standard deviation of the velocity vector in the extracted part ( $100 \times 100$  pixels) are calculated. Now, the cross-correlation algorithm is applied to the two images, blocks 1 and 2, to calculate a single displacement vector in the time interval between them.

(a) Zero-padded image at  $t_1$ .(b) Zero-padded image at  $t_2$ .

(c) Velocity field for zero-padded images.

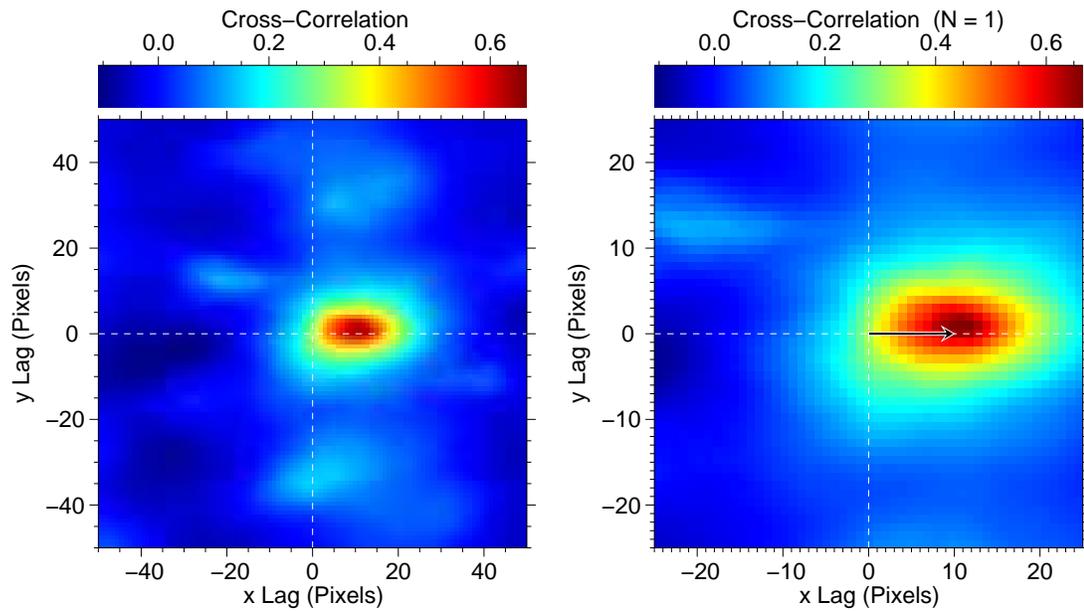
Figure 25. Panels a and b show  $100 \times 100$  subsets (blocks 1 and 2, respectively) that were extracted from images shown in Figure 24 and padded with zeros in preparation for cross-correlation. Panel c shows the corresponding velocity field for the  $100 \times 100$  pixel subset area.

### The Cross-Correlation Algorithm

Let  $f_1(x, y)$  and  $f_2(x, y)$  be the backscatter intensities of blocks 1 and 2, respectively. Then the cross-correlation function  $r_{x,y}$  for these images can be expressed as

$$r_{x,y} = F^{-1} \left[ \frac{F_1 F_2^*}{S_1 S_2} \right], \quad (7)$$

where  $F_1$  is the Fourier transform of  $f_1(x, y)$ ,  $F_2^*$  the complex conjugate of the Fourier transform of  $f_2(x, y)$ ,  $S_1$  is the standard deviation of  $f_1(x, y)$ ,  $S_2$  is the standard deviation of  $f_2(x, y)$ , and  $F^{-1}$  represents the inverse Fourier transform. The displacement  $(\Delta x, \Delta y)$  of block 2 can be estimated by the location of the peak of  $r_{x,y}$ . Figure 26 shows the cross-correlation function and the displacement vector (peak location of the cross-correlation function). From Figure 26, one can see that the location of the peak of the cross-correlation function indicates the horizontal component of the displacement.



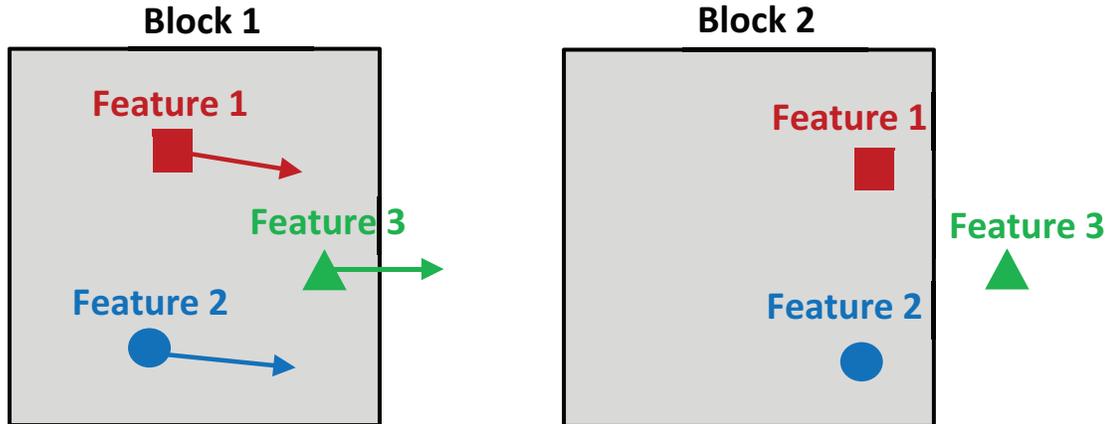
(a) Cross-correlation function.

(b) Displacement vector.

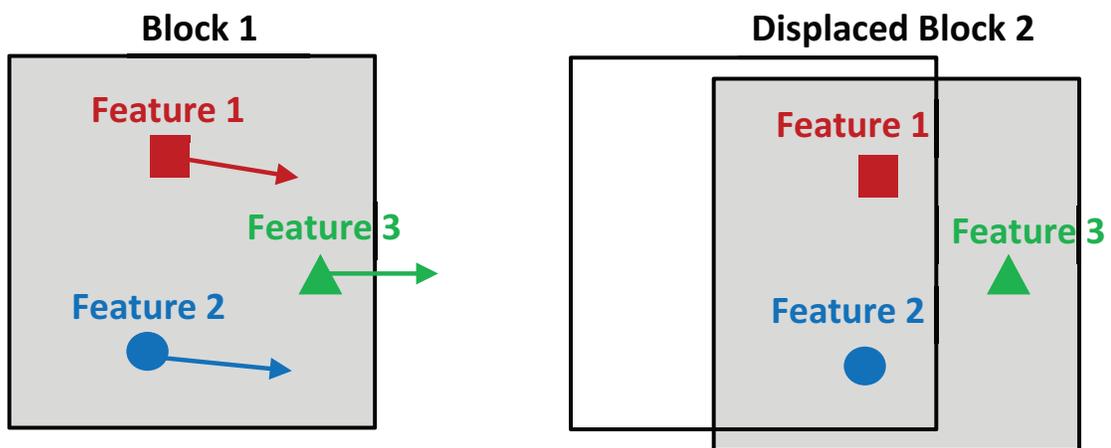
Figure 26. Cross-correlation function derived from blocks 1 and 2. The peak location of the cross-correlation function indicates the displacement as shown by the vector in panel b. Note: The cross-correlation shown in panel b is expanded to  $\pm 25$  pixels to show more detail near the peak.

### Multiple Pass Interrogation

The displacement vector, obtained from the peak location of the cross-correlation function, is based on features that appear in on both blocks 1 and 2. However, some features move out of the interrogation window in the time interval  $\Delta t = t_2 - t_1$  between these frames. Figure 27a represents blocks 1 and 2 in this situation. There are three features in block 1 (features 1, 2, and 3) and these features move at different velocities (shown with arrows in Figure 27a). During the time interval,  $\Delta t = t_2 - t_1$ , feature 3 moves out of the given interrogation window, and hence disappears in block 2, while the other two features remain in block 2. In this case, feature 3 does not contribute to the cross-correlation function of blocks 1 and 2, and the resultant displacement vector may be biased because of the disappearance of feature 3 in block 2. To resolve this problem, one applies “multiple pass interrogation” described by Raffel (2007). In this approach, the center of block 2 is displaced according to the displacement vector obtained from the peak location of the cross-correlation function (initial estimation). The displacement vector is found again using the cross-correlation function of block 1 and the displaced block 2 (correction of iterative refinement). Finally, by adding two displacements (initial estimation and correction of iterative method), one obtains the resultant displacement. Figure 27b shows the schematic of block 1 and displaced block 2 in this situation. From Figure 27b, one can see that feature 3 is in the displaced block 2 which contributes to finding the displacement vector. This technique allows the use of more data points and hence it increases the reliability of the displacement vector.



(a) Without multiple pass interrogation.

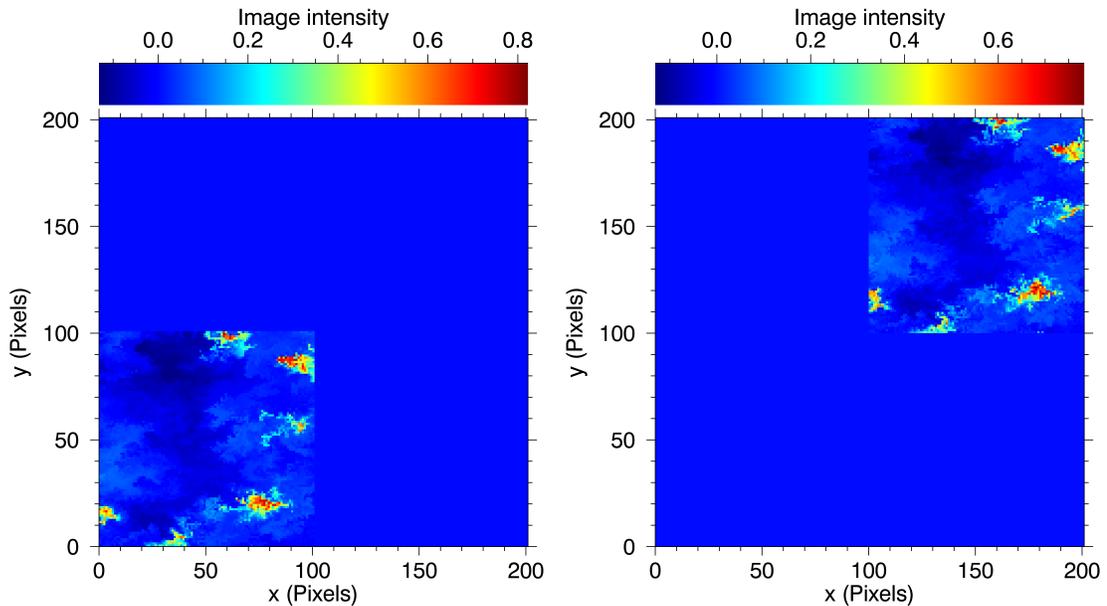


(b) With multiple pass interrogation.

Figure 27. Panel a shows blocks 1 and 2 without the multiple pass interrogation. In this case, feature 3 moves out in the time interval  $\Delta t = t_2 - t_1$ , and does not contribute to calculating the cross-correlation function. Panel b shows blocks 1 and 2 with multiple pass interrogation. In this case, Feature 3 appears in both block 1 and displaced block 2, and it contributes to the cross-correlation function.

The multiple pass interrogation has been used to estimate the wind velocity vectors by Mayor and Eloranta (2001), and is included in this investigation. Figures 28a and 28b show the synthetic backscatter images used for the approach. They are called “new block 1” and “new block 2,” respectively.

Figure 28a is identical to block 1 but Figure 28b is block 2 displaced by the initial estimation of the displacement vector. Using new blocks 1 and 2, one can find the correction of iterative refinement.



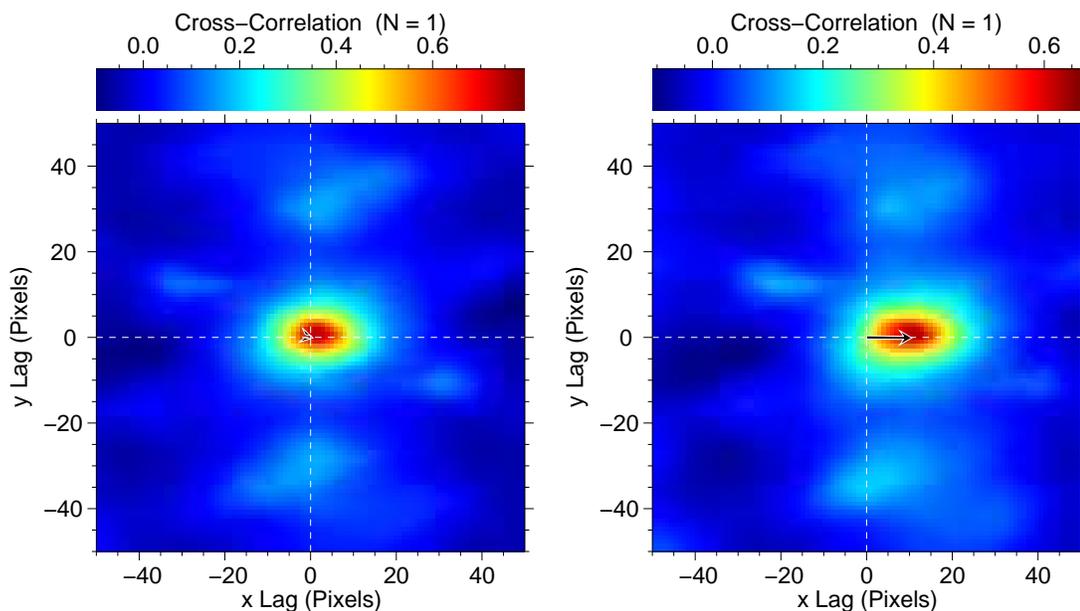
(a) Image at  $t_1$  for multi-pass approach. (b) Image at  $t_2$  for multi-pass approach.

Figure 28. Panel a shows the synthetic backscatter image at  $t_1$ , “new block 1”, used in the multi-pass approach. This image is exactly the same as block 1. Panel b shows the synthetic backscatter image at  $t_2$ , “new block 2”, used for multi-pass approach. This image is created by moving block 2 according to the estimated displacement vector resulting from the previous non-multi-pass cross-correlation function.

### Polynomial Fit to Obtain Subpixel Velocity Resolution

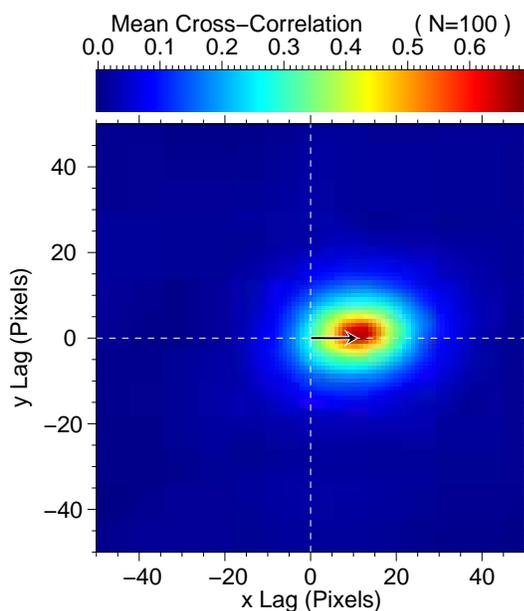
An improved velocity estimate can be obtained with the multipass approach described in the previous section. However, a limitation of resolution of the cross-correlation function remains. The peak location is given by the integer

value of pixels, and hence there is an uncertainty of  $\pm\frac{1}{2}$  pixels (Adrian and Westerweel, 2011). To reduce the uncertainty, one can use a polynomial fit of the cross-correlation function, and estimate the peak location with higher resolution, as described by Raffel (2007). Applying this technique, one may obtain subpixel resolution of the correction of iterative refinement, and the accuracy of the result may be significantly increased. Figure 29a shows the cross-correlation function and the correction of iterative refinement for the polynomial fitting of the cross-correlation function. Finally, the displacement vector is estimated by the sum of the initial estimation and the correction of iterative refinement. Figure 29b shows the cross-correlation function applied to blocks 1 and 2 with the final result of the displacement vector. This displacement vector depends on the distribution of the features in the interrogation window, so  $N = 100$  pairs of synthetic backscatter images (using different random seeds) are created and the mean and the standard deviation of the displacement vector are calculated. Figure 29c shows the mean cross-correlation function and the resultant displacement vector (mean displacement vector) for  $N = 100$ , synthetic backscatter images. By comparing Figures 29b and 29c, one can see that the cross-correlation function is smoothed and the peak is well defined if more synthetic backscatter images are used and the cross-correlation functions are averaged. Finally, the mean of the displacement vectors, for  $N = 100$  pairs of synthetic backscatter images, is compared with the mean of the wind velocity field described above.



(a) Correction of iterative refinement.

(b) Resultant displacement vector.



(c) Mean cross-correlation function.

Figure 29. Panel a shows the cross-correlation function derived from new blocks 1 and 2. The arrow represents the correction used in the iterative refinement. Panel b shows the cross-correlation function and the final displacement vector. The final displacement vector is the sum of the first estimated displacement, the refinement given by the multi-pass approach and the subpixel approximation. Panel c shows the mean cross-correlation function and the resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images.

### Effects of Turbulent Intensities

In the previous section, various 2-D velocity fields are used to test the performance of a cross-correlation algorithm. Let  $u(x, y)$  and  $v(x, y)$  be the east-west and the north-south components of the wind velocity  $(u, v)$ , respectively. Then,  $u(x, y)$  and  $v(x, y)$  can be expressed as

$$u(x, y) = u_0(x, y) + u'(x, y) \quad (8)$$

and

$$v(x, y) = v_0(x, y) + v'(x, y), \quad (9)$$

where  $u_0(x, y)$  and  $v_0(x, y)$  are analytical functions of  $x$  and  $y$ , and  $u'(x, y)$  and  $v'(x, y)$  are turbulent perturbations. To test the performance of the cross-correlation algorithm for environments without large coherent flow structures,  $u_0(x, y)$  and  $v_0(x, y)$  are held constant while increasing  $u'(x, y)$  and  $v'(x, y)$  in a series of experiments. In this case,  $u(x, y)$  and  $v(x, y)$  can be expressed as

$$u(x, y) = C_1 + u'(x, y) \quad (10)$$

and

$$v(x, y) = C_2 + v'(x, y), \quad (11)$$

where  $C_1$  and  $C_2$  are constants. Turbulent perturbations  $u'(x, y)$  and  $v'(x, y)$  are increased by increasing the aerodynamic roughness length  $z_0$ . According to Stull (1988), once the aerodynamic roughness length is determined for a particular surface, it does not change with wind speed, stability, or stress, but it can change if the roughness elements on the surface change. Table 1 shows the aerodynamic

roughness lengths for various terrains, and that the aerodynamic roughness length ranges from 0.0002 meters to a few meters.

TABLE 1  
THE AERODYNAMIC ROUGHNESS LENGTH

Terrain description	$z_0$ (m)
Open sea, Fetch at least 5 km	0.0002
Mud flats, snow; no vegetation, no obstacles	0.005
Open flat terrain; grass, few isolated obstacles	0.03
Low crops; occasional large obstacles	0.10
High crops; scattered obstacles	0.25
Parkland, bushes; numerous obstacles	0.5
Regular large obstacle coverage (suburb, forest)	1.0
City center with high- and low-rise buildings	$\geq 2$

The aerodynamic roughness length. Table taken from Davenport et al. (2000).

The magnitude of the turbulent perturbations of the wind velocity fields

$$(u, v) = (C_1 + u'(x, y), C_2 + v'(x, y)) \quad (12)$$

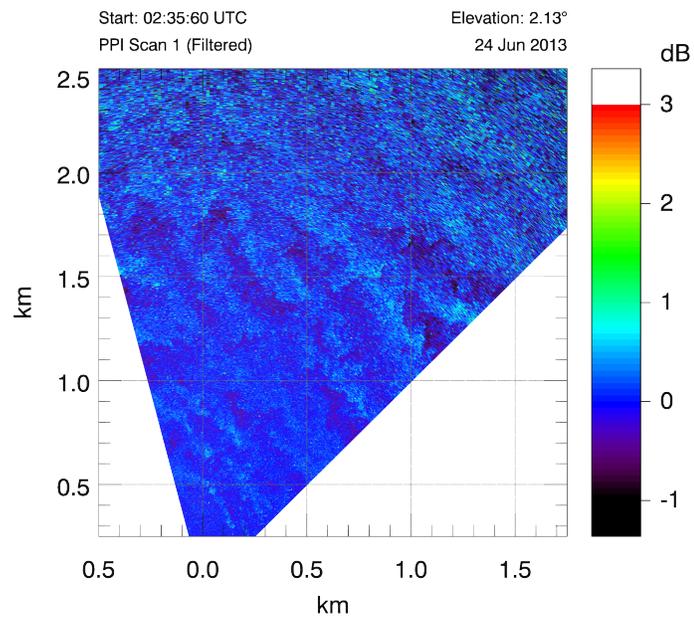
can be increased by increasing the aerodynamic roughness length  $z_0$ , and the performance of the cross-correlation algorithm is evaluated. As in the previous section,  $N = 100$  pairs of synthetic backscatter images are created, and the mean of the displacement vectors, obtained from the location of the peak of the cross-correlation function, is compared with the mean of the wind velocity field described above. Results are shown in the following chapter.

### Effects of the Edge of a Scan Sector

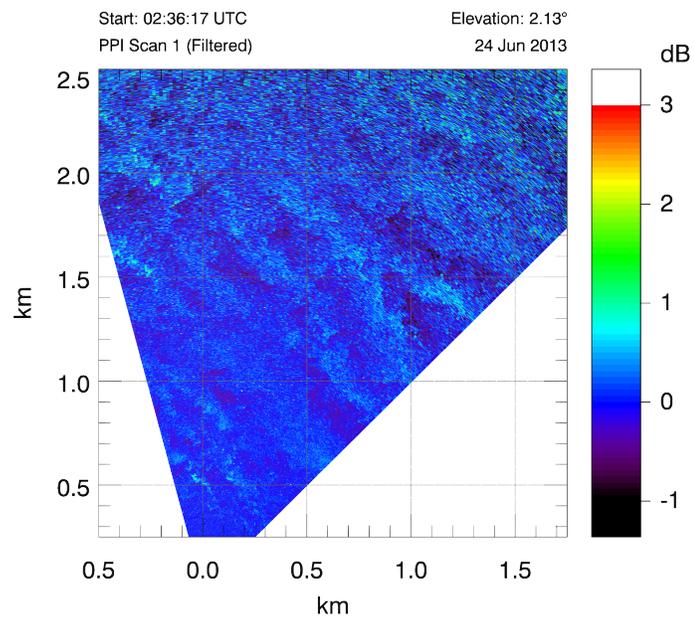
When aerosol features move across the edge of a lidar scan, the wind velocity estimation via cross-correlation algorithm is different from the expected value. Figures 30 and 31 show real lidar backscatter images on June 24, 2013 at California State University Chico, University Farm, when a bright aerosol feature passed through the right edge of the scan sector. Figure 32 shows the wind velocity fields estimated by applying the cross-correlation algorithm to the images in Figure 30. Here, the wind velocity vectors are generally directed from the southeast to the northwest. However, there is a strip of low velocity vectors near the right edge of the lidar scan. The effects are not noticeable near the left edge of the lidar scan where wind likely flows parallel to the edge line. To determine the cause of this apparent error synthetic backscatter images are cut in a way to mimic the edge of the sector scan.

The effects of an edge of a backscatter scan are tested in a way very similar to the one described in previous sections. However, instead of placing multiple Gaussian features randomly only one Gaussian feature is placed in the middle of the image. The intensity of the Gaussian feature is much greater than that of surrounding background features (about 10 times brighter). Next, turbulent diffusion is applied by using turbulent perturbation fields generated by the model of Mann (1994) and Mann (1998). Now, it is important to see what happens when the Gaussian feature is cut by an edge line. Thus, an edge line is calculated in such a way that the edge line cuts the Gaussian feature. Here, a diagonal edge line ( $y = x$ ) is used and half of the interrogation window, southeast side of the edge line, is filled in with zeros. Then, as in previous sections, one places this image ( $100 \times 100$  pixels) at the lower left corner of the interrogation window ( $200 \times 200$

pixels), and sets zeros for all other regions. Figure 33a show the synthetic backscatter image at time  $t_1$  (block 1) for testing the edge effect. Next, one applies a uniform velocity field to block 1 and obtains the synthetic backscatter image at time  $t_2$  (block 2). As in the previous sections, one places  $100 \times 100$  pixels of synthetic backscatter image and sets all zeros for the rest of region. However, the synthetic backscatter image is placed in the upper right of the interrogation window, instead of the lower left. Figures 33b and 33c show block 2 and the velocity field, respectively. Finally, the cross-correlation algorithm with multi-pass approach and polynomial fitting is applied, as described in the previous sections. The test is repeated by using  $N = 100$  pairs of synthetic backscatter images. One can place the same Gaussian feature in the middle of each synthetic backscatter image, but the distribution of surrounding features is varied.



(a) Lidar backscatter image at  $t_1$ .



(b) Lidar backscatter image at  $t_2$ .

Figure 30. REAL lidar scans on June 24, 2013 at California State University Chico, University Farm.

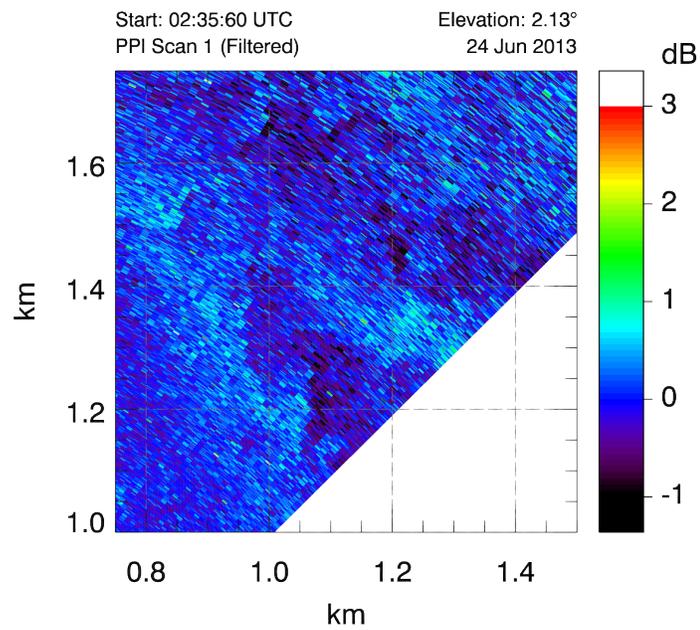
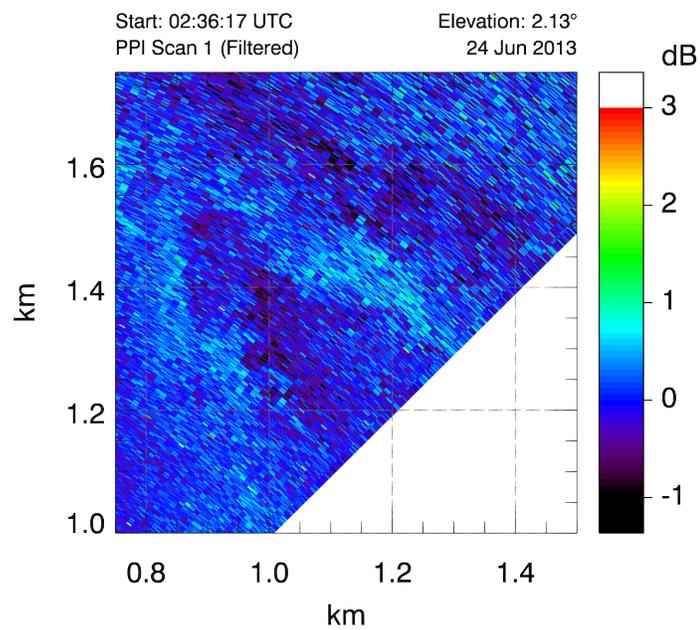
(a) Lidar backscatter image at  $t_1$ .(b) Lidar backscatter image at  $t_2$ .

Figure 31. Zoomed REAL lidar scans on June 24, 2013 at California State University Chico, University Farm.

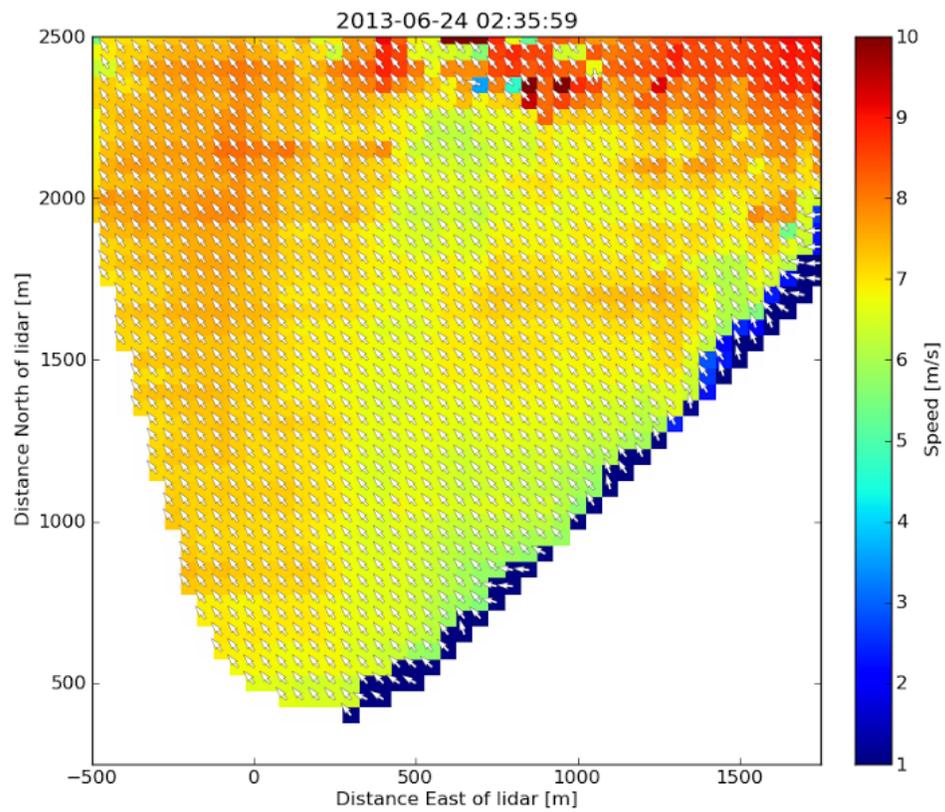
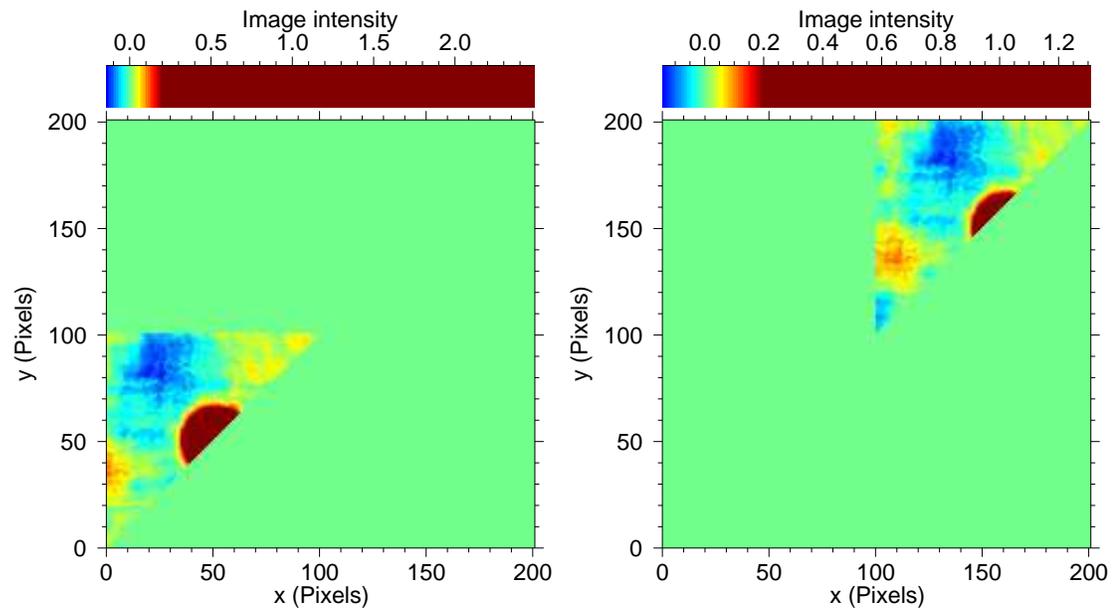
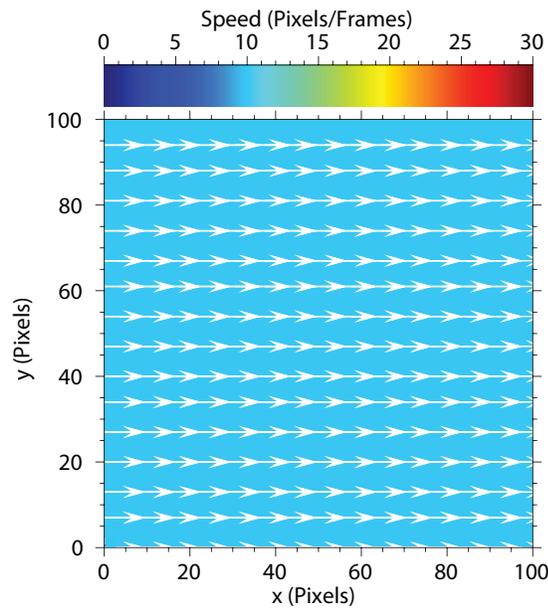


Figure 32. An example of the edge effect in a REAL lidar wind field. Wind velocity vectors were estimated by applying the cross-correlation algorithm on lidar backscatter images shown in Figures 30 and 31.



(a) Block 1 cut by an edge.

(b) Block 2 cut by an edge.



(c) Velocity field for testing edge effect.

Figure 33. Panel a shows block 1 with a Gaussian feature cut by an edge line. Panel b shows block 2 cut by an edge line. Block 2 is created by the velocity field, as shown in panel c, applied to block 1. Panel c is the uniform non-turbulent velocity field for testing the edge effect.

## CHAPTER IV

### RESULTS

The mean displacement vector, obtained from the cross-correlation algorithm  $N = 100$  pairs of synthetic backscatter images, is compared to the mean velocity for the corresponding synthetic velocity field. In this section, results of the following experiments are presented: (1) the performance of the cross-correlation algorithm for various velocity fields that are the sum of some analytical functions and turbulent perturbations, (2) the performance of the cross-correlation algorithm for varying turbulent intensities, and (3) the performance of the cross-correlation algorithm in the case that a predominant feature is cut by a edge line.

#### Different Types of Wind Velocity Fields

##### Uniform Flow

The uniform flow field is the velocity field such that the velocity in every pixel location in the interrogation window is constant. Let  $u(x, y)$  and  $v(x, y)$  be the east-west and the north-south components of the wind velocity, respectively, in the Cartesian coordinates  $(x, y)$ . Then the uniform flow field can be expressed as

$$(u, v) = (C_1, C_2), \tag{1}$$

where  $C_1$  and  $C_2$  are constants. One example of such velocity fields is shown in Figure 34a. Here, the velocity field is given by the function  $(u, v) = (10, 0)$

(pixels/frames), which is uniform and directed toward the east (positive x-direction). The colors and arrows in Figure 34a represent the magnitude and direction of the velocity vectors at a given pixel location, respectively. Since every pixel location in the interrogation window has the same velocity vector, the mean and the standard deviation of the velocity vectors are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 0.00)$  (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 34b shows the result of the cross-correlation algorithm for uniform flow  $(u, v) = (10.0, 0.00)$  (pixels/frames). The cross-correlation functions (functions of lags) for  $N = 100$  pairs of synthetic backscatter images are accumulated, and the mean value for each location is computed. The colors of the plot represent the mean cross-correlation values. The red and dark blue colors represent high and low mean cross-correlation values, respectively. From Figure 34b one can see that the peak of the cross-correlation function has coordinates  $(x_{lag}, y_{lag}) = (10.0, 0.00)$  (pixels), and represents the displacement in the time interval  $\Delta t = t_2 - t_1$ . For  $N = 100$  pairs of synthetic backscatter images, the mean and the standard deviation of the resultant displacement vector, calculated by the cross-correlation algorithm with multi-pass approach, are  $(9.98, 5.38 \times 10^{-5})$  (pixels) and  $(3.55 \times 10^{-3}, 4.44 \times 10^{-4})$  (pixels), respectively. The x-component of the mean displacement vector is 0.20% lower than that of the given velocity field. The resultant displacement vector is almost identical to that expected in the given velocity field. However, it is slightly smaller than the actual value, due to the fact that the cross-correlation function is always skewed (weighted toward the lower side) and the polynomial fit cannot give the exact peak location of the cross-correlation function.

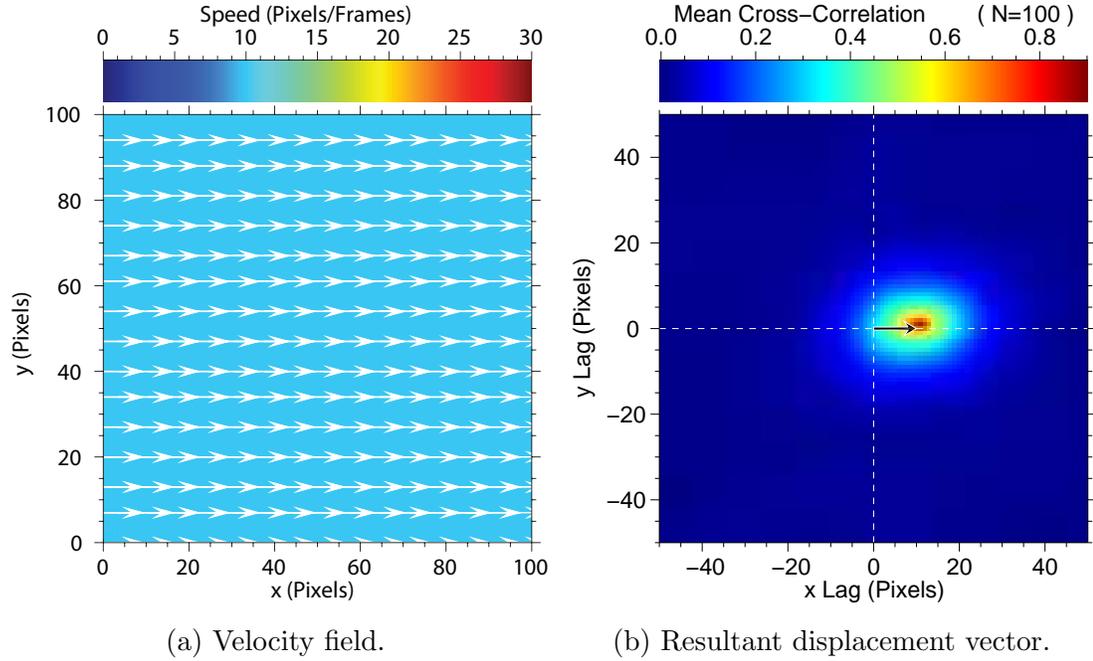


Figure 34. Panel a represents an example of a non-turbulent uniform velocity field,  $(u, v) = (10, 0)$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 0.00)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(9.98, 5.38 \times 10^{-5})$  (pixels) and  $(3.55 \times 10^{-3}, 4.44 \times 10^{-4})$  (pixels), respectively.

Uniform Flow with Turbulent  
Perturbations

Turbulent perturbations of wind velocity fields are common near the surface of the Earth. One simplest example of such field can be created by the sum of a uniform flow and a turbulent perturbation field, as generated by the model of Mann (1994) and Mann (1998). This type of flow is what may occur in flow over the ocean without large coherent structures. Let  $(u', v')$  represent the turbulent perturbation field. Then the velocity field,  $(u, v)$ , can be expressed as

$$(u, v) = (C_1 + u', C_2 + v'), \quad (2)$$

where,  $C_1$  and  $C_2$  are constants. One example of such a velocity field is,  $(u, v) = (10 + u', v')$  (pixels/frames), as shown in Figure 35a. Panel a shows that the velocity field is not constant because of the turbulent perturbations, but most of the velocity vectors are directed to the right (positive x direction). The mean and the standard deviation of the velocity vectors are (11.1, 0.104) (pixels/frames) and (1.97, 1.23) (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Panel b shows the result of the cross-correlation algorithm for the uniform flow with turbulent fluctuations. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are (10.8, 0.0702) (pixels) and (0.778, 0.289) (pixels), respectively. The x-component of the mean displacement vector is 2.70% lower than that of the given velocity field. The resultant displacement vectors are slightly lower than the displacement vector predicted by the given velocity field.

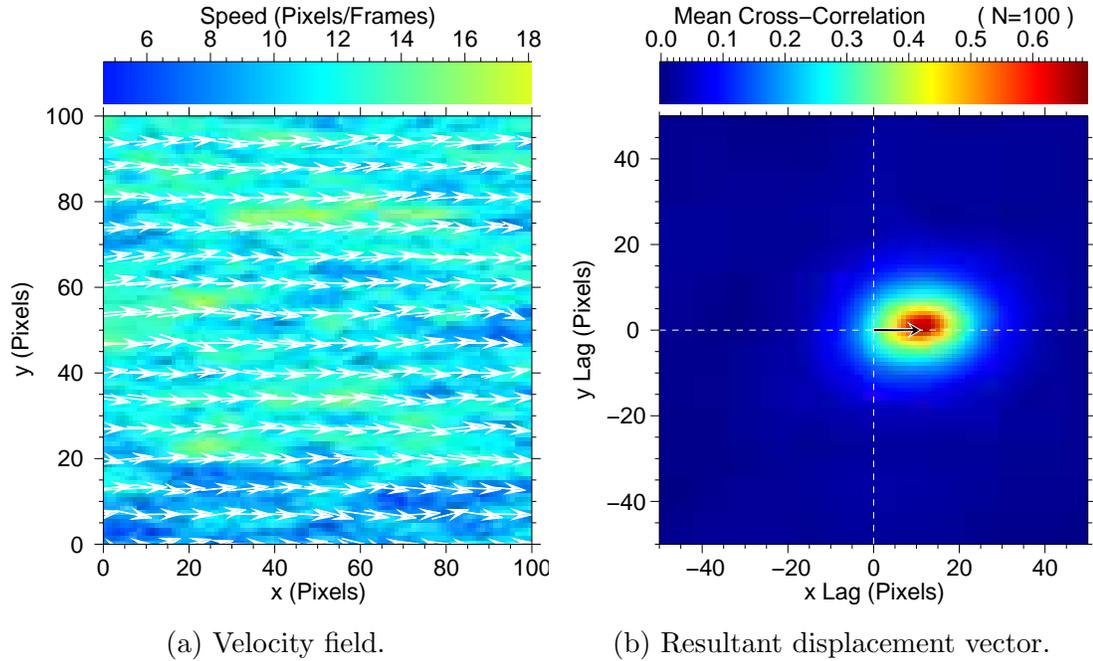


Figure 35. Panel a represents an example of a uniform velocity field with turbulent perturbations,  $(u, v) = (10 + u', v')$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(11.1, 0.104)$  (pixels/frames) and  $(1.97, 1.23)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(10.8, 0.0702)$  (pixels) and  $(0.778, 0.289)$  (pixels), respectively.

### Converging Flow

Convergent flow on the scale of a typical interrogation window (100 m - 1000 m) occurs routinely in unstable daytime convective boundary layers. Areas of strong horizontal convergence and divergence are associated with cellular convection circulations in the lower boundary layer (Schmidt and Schumann, 1989). The convergent flow field  $(u, v)$  has a negative divergence, so that

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} < 0. \quad (3)$$

The notion of convergence requires 2 components  $u$  and  $v$ . One example of velocity fields that satisfies the equation above is  $(u, v) = (10, -0.2y + 10)$  (pixels/frames), applied to the  $xy$ -plane ( $0 \leq x \leq 100, 0 \leq y \leq 100$ ), as shown in Figure 36a. Here, the flow field is constant in the  $u$ -component, but there is a convergence in the  $v$ -component. The mean and the standard deviation of the velocity vectors are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 5.83)$  (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 36b shows the result of the cross-correlation algorithm for the convergent flow. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are  $(9.84, -0.533)$  (pixels) and  $(0.896, 3.85)$  (pixels), respectively. The  $x$ -component of the mean displacement vector is 1.60% lower than that of the given velocity field. The resultant displacement vectors are slightly lower than the displacement vector predicted by the given velocity field, and the standard deviation in the  $y$ -direction is higher than in the previous cases because of non-uniformity of the velocity field in the  $y$ -direction.

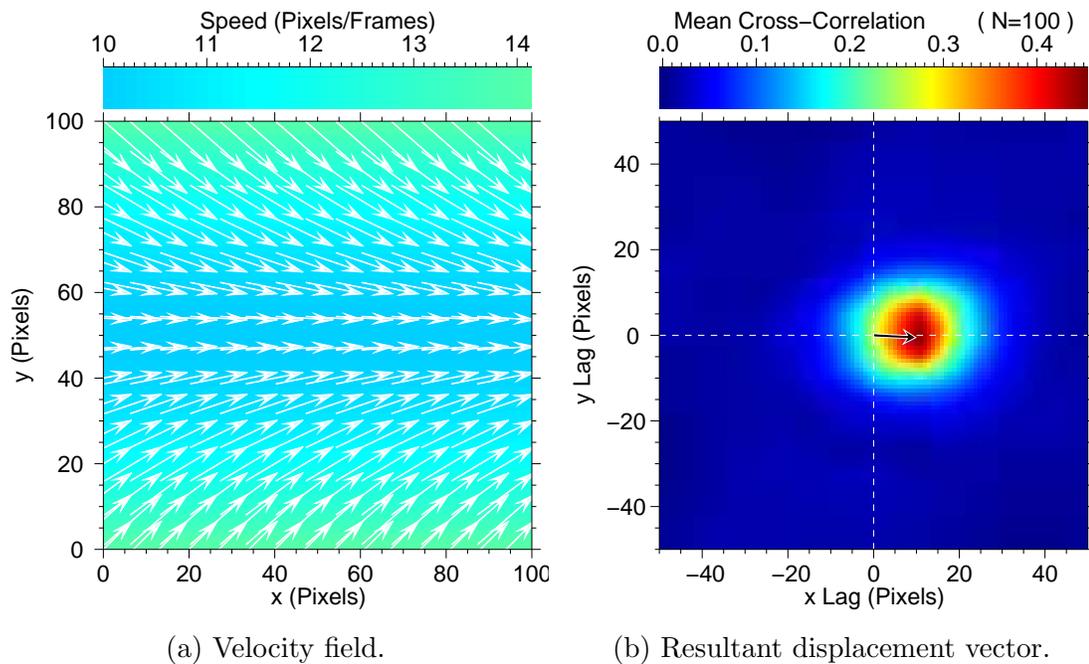


Figure 36. Panel a represents an example of a non-turbulent convergent velocity field,  $(u, v) = (10, -0.2y + 10)$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 5.83)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(9.84, -0.533)$  (pixels) and  $(0.896, 3.85)$  (pixels), respectively.

### Converging Flow with Turbulent Perturbations

The turbulent perturbation field, as generated by the model of Mann (1994) and Mann (1998), is added to the convergent flow field to examine non-uniform converging velocity fields. One example of a velocity field is  $(u, v) = (10 + u', -0.2y + 10 + v')$  (pixels/frames), where  $u'$  and  $v'$  are the u- and v-components of the turbulent perturbations, respectively. Figure 37a shows the convergent flow with turbulent perturbations  $(u, v) = (10 + u', -0.2y + 10 + v')$  (pixels/frames), applied to the xy-plane ( $0 \leq x \leq 100, 0 \leq y \leq 100$ ). The mean and the standard deviation of the velocity vectors are  $(11.1, 0.103)$  (pixels/frames) and  $(1.97, 5.73)$  (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 37b shows the result of the cross-correlation algorithm for the convergent flow with turbulent perturbations. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are  $(10.7, -0.395)$  (pixels) and  $(1.55, 3.54)$  (pixels), respectively. The x-component of the mean displacement vector is 3.60% lower than that of the given velocity field. The resultant displacement vectors are slightly lower than the displacement vector predicted by the given velocity field, and there are standard deviations in both x- and y-directions due to non-uniformity of the velocity field in the interrogation window.

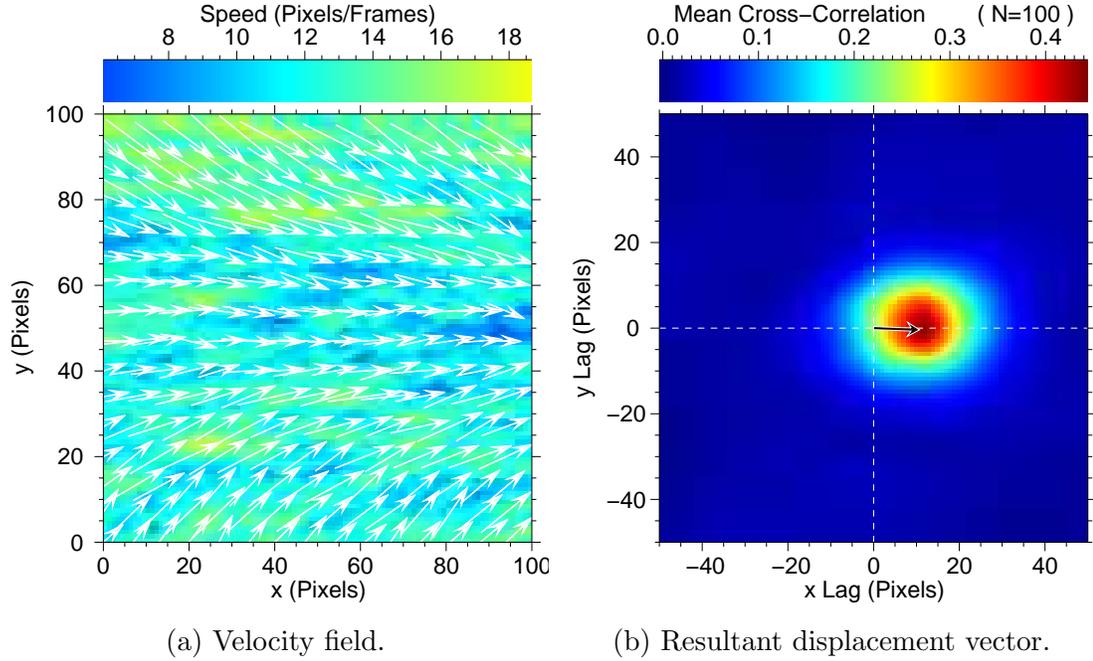


Figure 37. Panel a represents an example of a convergent velocity field with turbulent perturbations,  $(10 + u', -0.2y + 10 + v')$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(11.1, 0.103)$  (pixels/frames) and  $(1.97, 5.73)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(10.7, -0.395)$  (pixels) and  $(1.55, 3.54)$  (pixels), respectively.

### Diverging Flow

Divergent flow on the scale of a typical interrogation window (100 m - 1000 m) occurs routinely in unstable daytime convective boundary layers. Areas of strong horizontal convergence and divergence are associated with cellular convection circulations in the lower boundary layer (Schmidt and Schumann, 1989). The divergent flow field  $(u, v)$  has a positive divergence, so that

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} > 0. \quad (4)$$

The notion of divergence requires 2 components  $u$  and  $v$ . One example of velocity fields that satisfies the equation above is  $(u, v) = (10, 0.2y - 10)$  (pixels/frames), applied to the  $xy$ -plane ( $0 \leq x \leq 100, 0 \leq y \leq 100$ ), as shown in Figure 38a. As in the convergent flow described above, the flow field is constant in the  $u$ -component, but there is a divergence in the  $v$ -component. The mean and the standard deviation of the velocity vectors are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 5.83)$  (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 38b shows the result of the cross-correlation algorithm for the convergent flow. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are  $(9.89, 0.503)$  (pixels) and  $(0.484, 5.35)$  (pixels), respectively. The  $x$ -component of the mean displacement vector is 1.10% lower than that of the given velocity field. The resultant displacement vectors are slightly lower than the displacement vector predicted by the given velocity field, as in the case of the convergent flow.

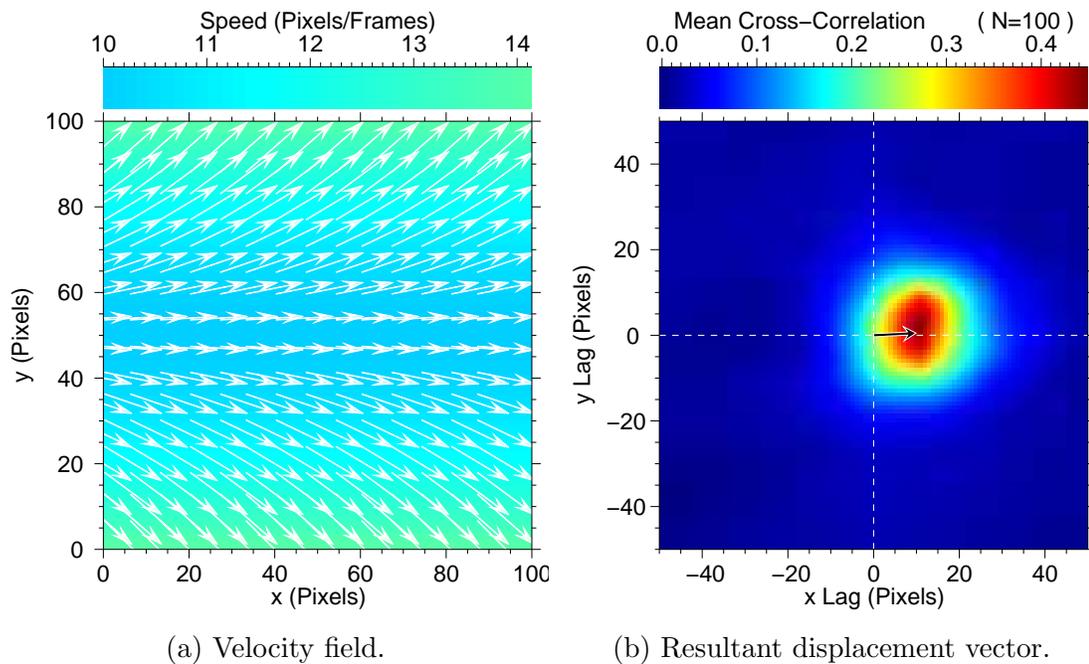


Figure 38. Panel a represents an example of a non-turbulent divergent velocity field,  $(u, v) = (10, 0.2y - 10)$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 5.83)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(9.89, 0.533)$  (pixels) and  $(0.484, 5.35)$  (pixels), respectively.

### Diverging Flow with Turbulent Perturbations

The turbulent perturbation field, as generated by the model of Mann (1994) and Mann (1998), is added to the divergent flow field to examine non-uniform divergent velocity fields. One example of such velocity field is  $(u, v) = (10 + u', 0.2y - 10 + v')$  (pixels/frames), where  $u'$  and  $v'$  are the u- and v-components of the turbulent perturbations, respectively. Figure 39a shows the divergent flow with turbulent perturbations  $(u, v) = (10 + u', 0.2y - 10 + v')$  (pixels/frames), applied to the xy-plane ( $0 \leq x \leq 100, 0 \leq y \leq 100$ ). The mean and the standard deviation of the velocity vectors are  $(11.1, 0.103)$  (pixels/frames) and  $(1.97, 6.18)$  (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 39b shows the result of the cross-correlation algorithm for the divergent flow with turbulent perturbations. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are  $(10.6, 0.553)$  (pixels) and  $(1.72, 5.35)$  (pixels), respectively. The x-component of the mean displacement vector is 4.50% lower than that of the given velocity field. As in the convergent flow with turbulent perturbations case, the resultant displacement vectors are slightly lower than the displacement vector predicted by the given velocity field, and there are standard deviations in both x- and y-directions due to non-uniformity of the velocity field in the interrogation window. From these results, one can say that the resultant displacement vectors are similar for divergent and convergent flow fields.

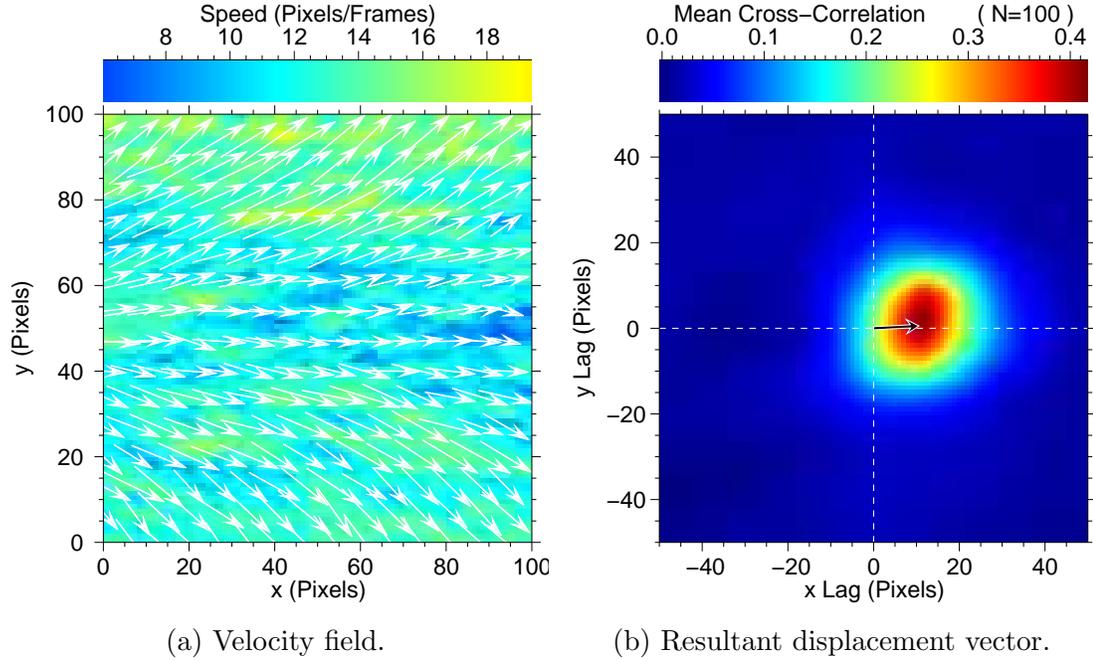


Figure 39. Panel a represents an example of a divergent velocity field with turbulent perturbations,  $(10+u', 0.2y-10+v')$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(11.1, 0.103)$  (pixels/frames) and  $(1.97, 6.18)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(10.6, 0.553)$  (pixels) and  $(1.72, 5.35)$  (pixels), respectively.

### Rotational Flow

Rotational flow on the scale of a typical interrogation window (100 m - 1000 m) occurs routinely in unstable daytime convective boundary layers. The rotation is often observed to occur at the intersection of convergence boundaries (Kanak, 2005). The circular flow  $(u, v)$  in the  $xy$ -plane, needs to satisfy the following condition.

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \neq 0. \quad (5)$$

If the velocity field is in the counterclockwise direction (viewed from the top of the  $xy$ -plane), the velocity field  $(u, v)$  satisfies the condition

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} > 0 \quad (6)$$

or, if the velocity field is in the clockwise direction, the velocity field  $(u, v)$  satisfies the condition

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} < 0. \quad (7)$$

One example of rotational flow in the counterclockwise direction is

$(u, v) = (-0.2y + 10, 0.2x - 10)$  (pixels/frames), applied to the  $xy$ -plane ( $0 \leq x \leq 100, 0 \leq y \leq 100$ ), as shown in Figure 40a. From Figure 40a, one can see that the flow field has a rotation and the  $u$ - and  $v$ -components of the fields have both positive and negative values in the interrogation window, and are canceled out. The mean and the standard deviation of the velocity vectors are  $(5.98 \times 10^{-5}, 0.00)$  (pixels/frames) and  $(5.83, 5.83)$  (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 40b shows the result of the cross-correlation algorithm for the rotational flow. The mean and the standard deviation of the resultant displacement vectors,

due to different synthetic backscatter images, are  $(-0.891, -0.332)$  (pixels) and  $(4.19, 4.74)$  (pixels), respectively. The difference between x- and y- components of the mean displacement vector and that of actual flow field is within  $\pm 1$  pixel. The resultant displacement vectors depend on the distribution of features in the image, so the standard deviations are high in x- and y-components. However, the mean cross-correlation function averages and cancels fluctuations of the resultant displacement vectors, and gives a low mean displacement vector that is somewhat similar to the one predicted from the given rotational flow field.

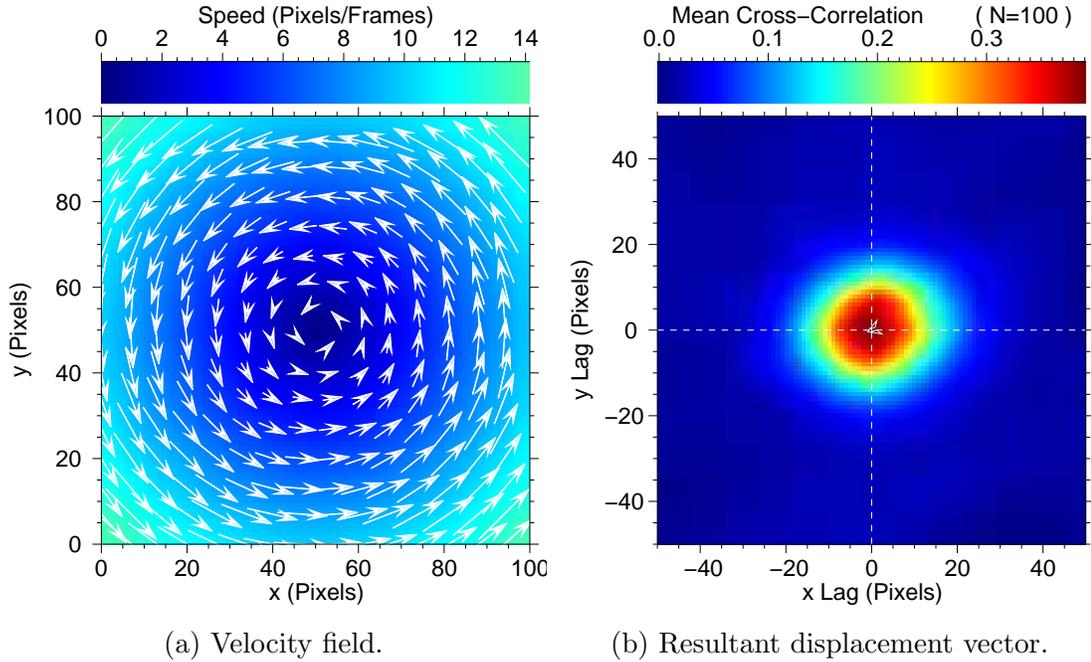


Figure 40. Panel a represents an example of a non-turbulent rotational velocity field,  $(u, v) = (-0.2y + 10, 0.2x - 10)$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(5.98 \times 10^{-5}, 0.00)$  (pixels/frames) and  $(5.83, 5.83)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(-0.891, -0.332)$  (pixels) and  $(4.19, 4.74)$  (pixels), respectively.

Rotational Flow with Turbulent  
Perturbations

The turbulent perturbation field, as generated by the model of Mann (1994) and Mann (1998), is added to the rotational flow field to examine non-uniform rotational velocity fields. One example of such velocity field is  $(u, v) = (-0.2y + 10 + u', 0.2x - 10 + v')$  (pixels/frames), where  $u'$  and  $v'$  are the u- and v- components of the turbulent perturbations, respectively. Figure 41a shows the rotational flow with turbulent perturbations  $(u, v) = (-0.2y + 10 + u', 0.2x - 10 + v')$  (pixels/frames), applied to the xy-plane ( $0 \leq x \leq 100, 0 \leq y \leq 100$ ). The mean and the standard deviation of the velocity vectors are (1.08, 0.104) (pixels/frames) and (5.46, 5.92) (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 41b shows the result of the cross-correlation algorithm for the rotational flow with turbulent perturbations. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are (0.345, -0.118) (pixels) and (3.38, 4.34) (pixels), respectively. As the case of rotational flow with no turbulent perturbations, the difference between x- and y-components of the mean displacement vector and that of the actual flow field is within  $\pm 1$  pixel. The results are similar to those of the rotational flow with no turbulent perturbations.

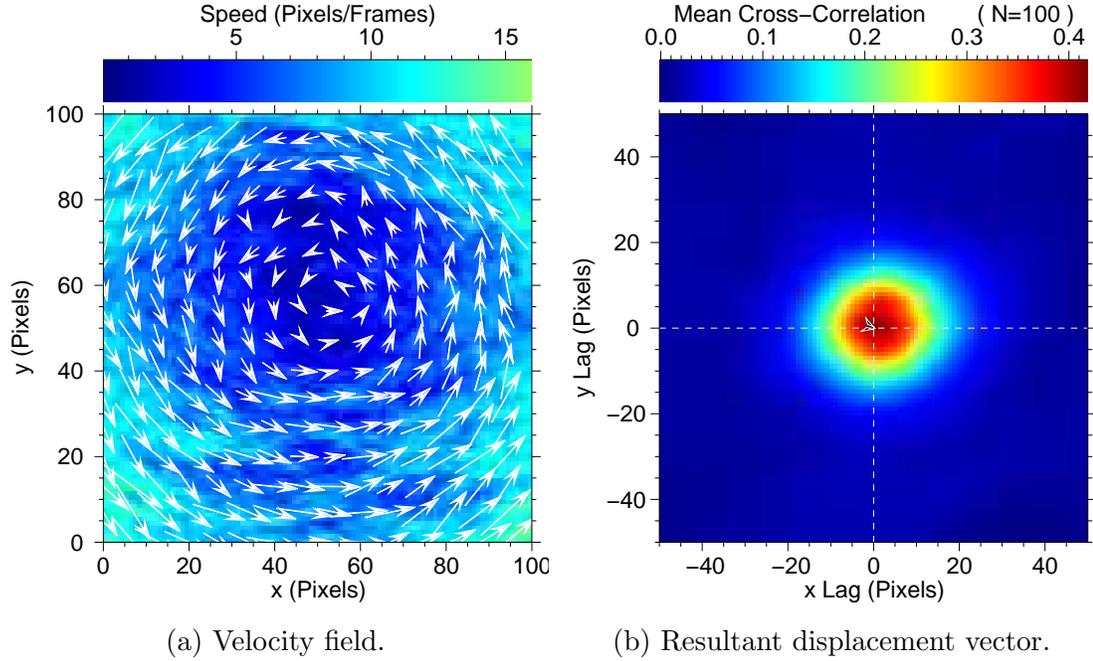


Figure 41. Panel a represents an example of a rotational velocity field with turbulent perturbations,  $(u, v) = (-0.2y + 10 + u', 0.2x - 10 + v')$  (pixels/frames), applied to block 1. The mean velocity and standard deviation are  $(1.08, 0.104)$  (pixels/frames) and  $(5.46, 5.92)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(0.345, -0.118)$  (pixels) and  $(3.38, 4.34)$  (pixels), respectively.

### Shear Flow

Concentrated regions of horizontal wind shear may occur in the lower atmosphere and are associated with a variety of mesoscale flows. Some examples of such shear flows are density current fronts (Mayor, 2011) and flow over complex terrain. The shear flow is considered as the extreme case of the non-uniform flow where two sections within the interrogation window have opposing velocities. One example of such velocity field is created by a constant negative velocity  $(u, v) = (-5, 0)$  (pixels/frames) in the upper part and a constant positive velocity  $(u, v) = (25, 0)$  (pixels/frames) in the lower part, joined by a shear profile, as shown in Figure 42a. Since the lower part of the velocity is greater than that of the upper part, the mean velocity in the interrogation window is in the positive u-direction. The mean and the standard deviation of the velocity vectors are  $(10.2, 0.00)$  (pixels/frames) and  $(14.3, 0.00)$  (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 42b shows the result of the cross-correlation algorithm for the shear flow with turbulent perturbations. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are  $(4.08, -0.0860)$  (pixels) and  $(15.3, 0.775)$  (pixels), respectively. The difference between the x-component of the mean displacement vector and that of actual flow field is about 6 pixels, and it clearly underestimates the given flow field. From Figure 42b we see that the mean cross-correlation function has two distinct peaks  $(x_{lag}, y_{lag}) = (25, 0)$  (pixels) and  $(x_{lag}, y_{lag}) = (-5, 0)$  (pixels), because of the distribution of features. If the predominant features are in the upper part of the image, the cross-correlation algorithm gives the displacement vector  $(x, y) = (-5, 0)$  (pixels). On the other hand, if predominant features are in the lower part of the image, the

cross-correlation algorithm gives the displacement vector  $(x, y) = (25, 0)$  (pixels). These displacement vectors are averaged, but the cross-correlation algorithm tends to pick more lower displacement vectors, so the resultant displacement vectors are much lower than the one predicted from the given velocity field.

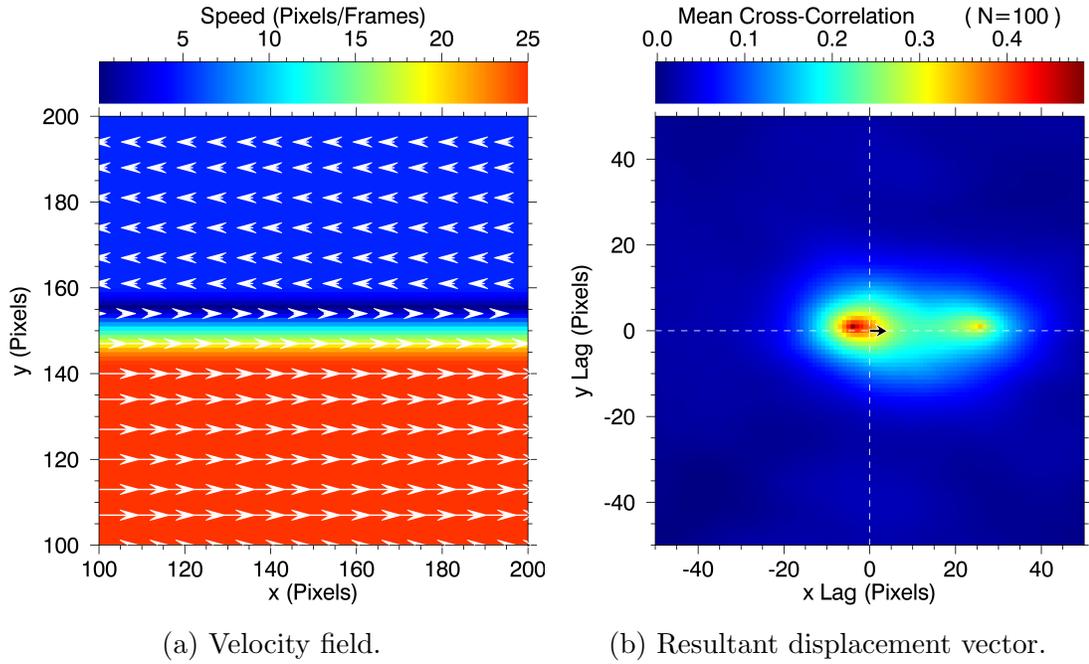


Figure 42. Panel a represents an example of non-turbulent velocity field with strong shear applied to block 1. The velocity field is created by a constant negative velocity  $(u, v) = (-5, 0)$  (pixels/frames) in the upper part and a constant positive velocity  $(u, v) = (25, 0)$  (pixels/frames) in the lower part, jointed by a shear profile. The mean velocity and standard deviation are  $(10.2, 0.00)$  (pixels/frames) and  $(14.3, 0.00)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(4.08, -0.0860)$  (pixels) and  $(15.3, 0.775)$  (pixels), respectively.

### Shear Flow with Turbulent Perturbations

Finally, the turbulent perturbation field, as generated by the model of Mann (1994) and Mann (1998), is added to the shear flow field. One example of such velocity field is the turbulent perturbation field ( $u'v'$ ) added to the shear field in Figure 42a. Figure 43a shows the shear flow with turbulent perturbations applied to the xy-plane ( $0 \leq x \leq 100, 0 \leq y \leq 100$ ). The mean and the standard deviation of the velocity vectors are (11.2, 0.104) (pixels/frames) and (13.9, 1.23) (pixels/frames), respectively. This velocity field is applied to  $N = 100$  pairs of synthetic backscatter images. Figure 43b shows the result of the cross-correlation algorithm for the shear flow with turbulent perturbations. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are (4.95, -0.115) (pixels) and (14.2, 0.978) (pixels), respectively. The difference between the x-component of the mean displacement vector and that of actual flow field is about 6 pixels, and it clearly underestimates the given flow field. The results are similar to that of the shear flow with no turbulent perturbations.

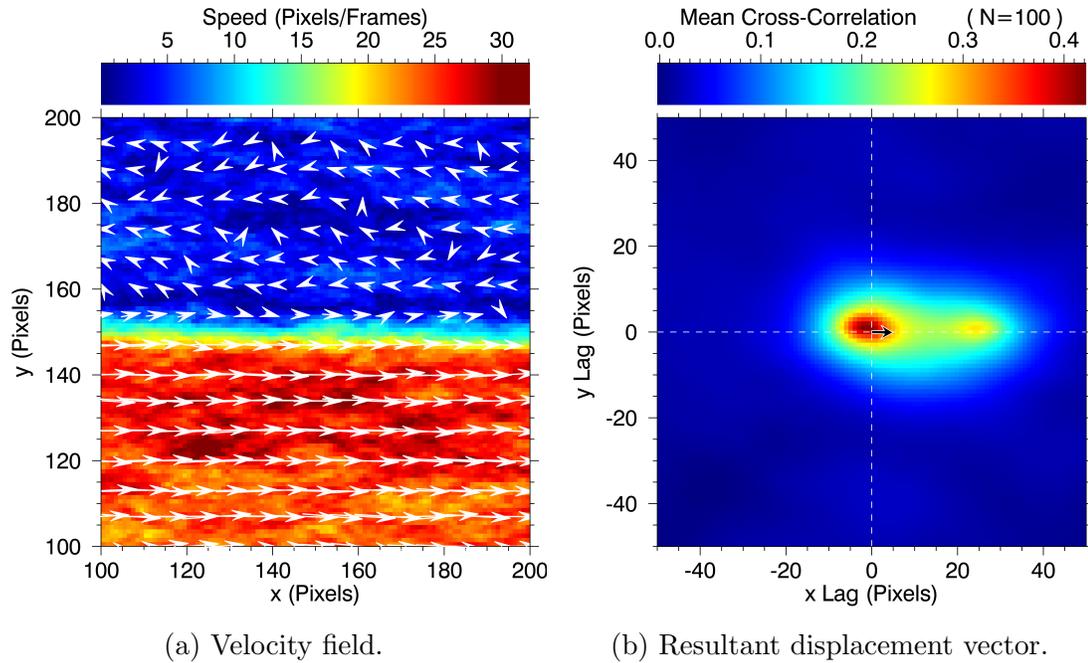


Figure 43. Panel a represents an example of a shear velocity field with turbulent perturbations applied to block 1. The velocity field is created by the sum of shear velocity field, as shown in Figure 42a, and turbulent perturbations. The mean velocity and standard deviation are  $(11.2, 0.104)$  (pixels/frames) and  $(13.9, 1.23)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(4.95, -0.115)$  (pixels) and  $(14.2, 0.978)$  (pixels), respectively.

### Effects of Turbulent Intensities

The relationship between non-uniformity of velocity fields and the performance of the cross-correlation algorithm can be examined by using the velocity field

$$(u, v) = (C_1 + u'(x, y), C_2 + v'(x, y)), \quad (8)$$

where  $u'(x, y)$  and  $v'(x, y)$  are turbulent perturbations and  $C_1$  and  $C_2$  are constants.  $u'(x, y)$  and  $v'(x, y)$  are increased by increasing the aerodynamic roughness length  $z_0$  for a series of flow fields. One example of such velocity field is

$$(u, v) = (10 + u'(x, y), v'(x, y)) \quad (9)$$

with different values of  $z_0$ , as shown in Figure 44 and Table 2. Figure 44 and Table 2 show that the velocity field is approximately uniform in the interrogation window for  $z_0 = 0.001$  m, while turbulent perturbations are noticeable for large values of  $z_0$ .

Figure 45 shows mean cross-correlation functions for the velocity field  $(u, v) = (10 + u'(x, y), v'(x, y))$  (pixels/frames) with different values of the aerodynamic roughness length  $z_0$ . From Figure 45, one can see that the mean cross-correlation function is broadened as the aerodynamic roughness length increases. Table 3 shows the mean and standard deviation of resultant displacement vectors for  $N = 100$  pairs of synthetic backscatter images. A comparison of Table 2 and Table 3 shows that the cross-correlation algorithm tends to underestimate the velocity and underestimation is more noticeable when the aerodynamic roughness length  $z_0$  is increased.

Let  $D_{true}$  represent the magnitude of the mean displacement vector obtained from wind velocity field, and  $D_{ccf}$  represent the magnitude of the displacement vector from the cross-correlation function  $D_{ccf}$  using  $N = 100$  pairs of synthetic

TABLE 2  
VELOCITY FIELDS WITH DIFFERENT AERODYNAMIC ROUGHNESS  
LENGTHS  $z_0$

$z_0$ (m)	Mean Velocity ( $\bar{u}, \bar{v}$ )	Standard Deviation
0.0001	( 10.57, 0.047 )	( 0.647, 0.376 )
0.0005	( 10.64, 0.053 )	( 0.732, 0.425 )
0.001	( 10.68, 0.056 )	( 0.776, 0.451 )
0.005	( 10.79, 0.065 )	( 0.902, 0.524 )
0.01	( 10.85, 0.070 )	( 0.970, 0.564 )
0.05	( 11.03, 0.085 )	( 1.18, 0.683 )
0.1	( 11.13, 0.094 )	( 1.29, 0.752 )
0.5	( 11.48, 0.123 )	( 1.69, 0.980 )
1.0	( 11.70, 0.141 )	( 1.94, 1.13 )

Velocity fields  $(u, v) = (10 + u', v')$  (pixels/frames) with different aerodynamic roughness length  $z_0$ . The mean and standard deviation are measured in (pixels/ frames). Turbulent fluctuation is increased as increasing  $z_0$ .

TABLE 3  
RESULTANT DISPLACEMENT VECTORS FOR  $N = 100$  PAIRS OF  
SYNTHETIC BACKSCATTER IMAGES

$z_0$ (m)	Mean Displacement ( $\bar{x}, \bar{y}$ )	Standard Deviation
0.0001	( 10.33, 0.0252 )	( 0.432, 0.155 )
0.0005	( 10.39, 0.0282 )	( 0.446, 0.159 )
0.001	( 10.42, 0.0277 )	( 0.449, 0.163 )
0.005	( 10.51, 0.0311 )	( 0.448, 0.158 )
0.01	( 10.54, 0.0371 )	( 0.461, 0.170 )
0.05	( 10.67, 0.0519 )	( 0.501, 0.191 )
0.1	( 10.73, 0.0717 )	( 0.527, 0.191 )
0.5	( 10.90, 0.117 )	( 0.740, 0.254 )
1.0	( 10.93, 0.137 )	( 0.856, 0.314 )

Resultant displacement vectors for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation are measured in (pixels/ frames).

backscatter images. Then the relative error and the absolute error can be expressed as  $|\frac{D_{ccf}-D_{true}}{D_{true}}|$  and  $|D_{ccf} - D_{true}|$ , respectively. Figure 46 shows the relative error  $|\frac{D_{ccf}-D_{true}}{D_{true}}|$  versus the aerodynamic roughness length  $z_0$  and the absolute error  $|D_{ccf} - D_{true}|$  versus the aerodynamic roughness length  $z_0$ . From Figure 46, one can see that both the relative error and the absolute error increase as the aerodynamic roughness length increases.

For a given turbulent perturbation field  $(u'(x, y), v'(x, y))$ , the turbulence kinetic energy (TKE) is defined as

$$TKE = \frac{1}{2} [\overline{u'^2} + \overline{v'^2}]. \quad (10)$$

Figure 47 shows the relative error  $|\frac{D_{ccf}-D_{true}}{D_{true}}|$  versus TKE and the absolute error  $|D_{ccf} - D_{true}|$  versus TKE. From Figure 47, one can see that both the relative error and the absolute error increase as TKE increases.

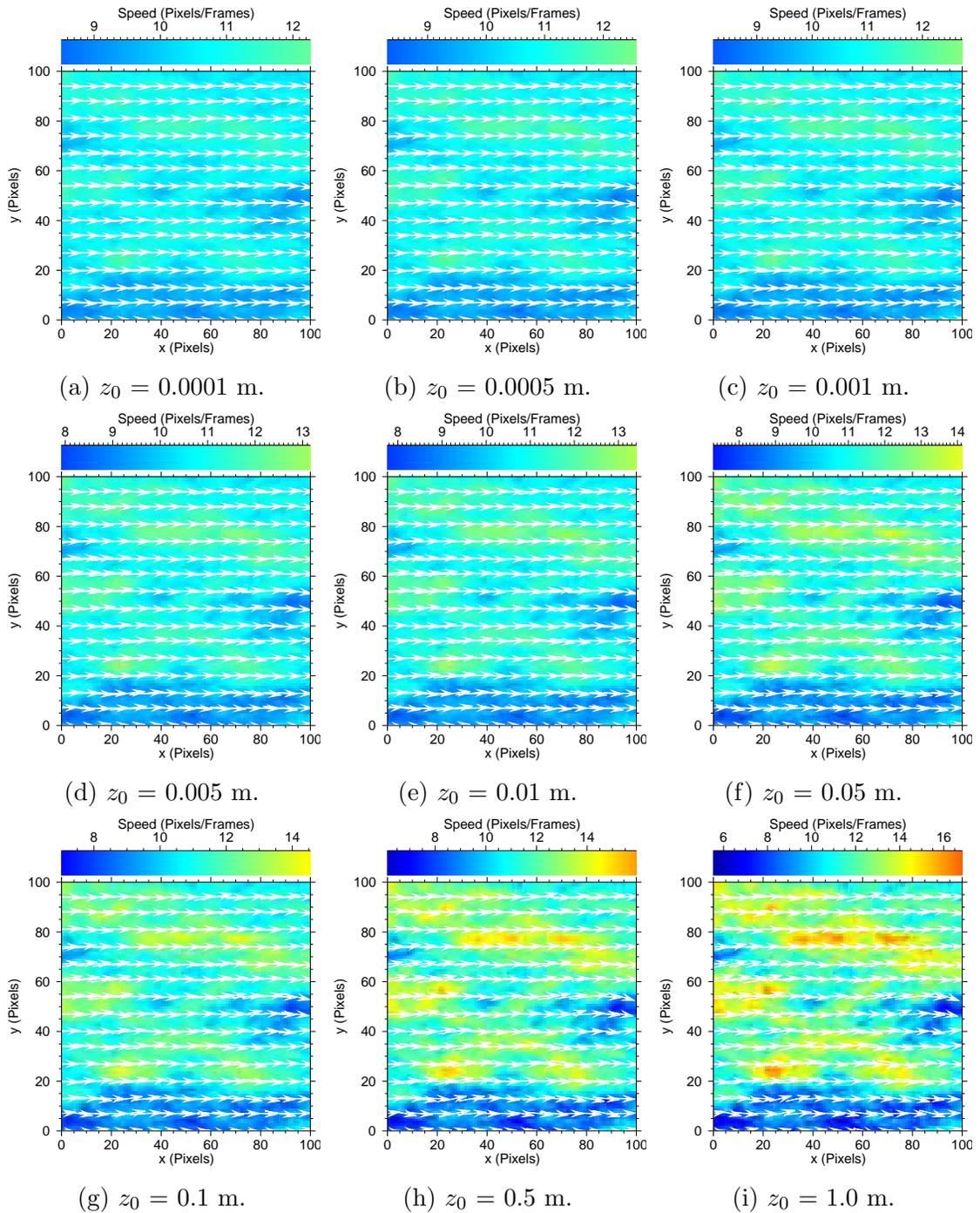


Figure 44. Velocity fields  $(u, v) = (10 + u'(x, y), v'(x, y))$  (pixels/frames) generated by uniform flow and various turbulent intensities. Turbulent intensities are increased by increasing the aerodynamic roughness length  $z_0$ .  $z_0$  changes from 0.0001 m to 1.0 m.

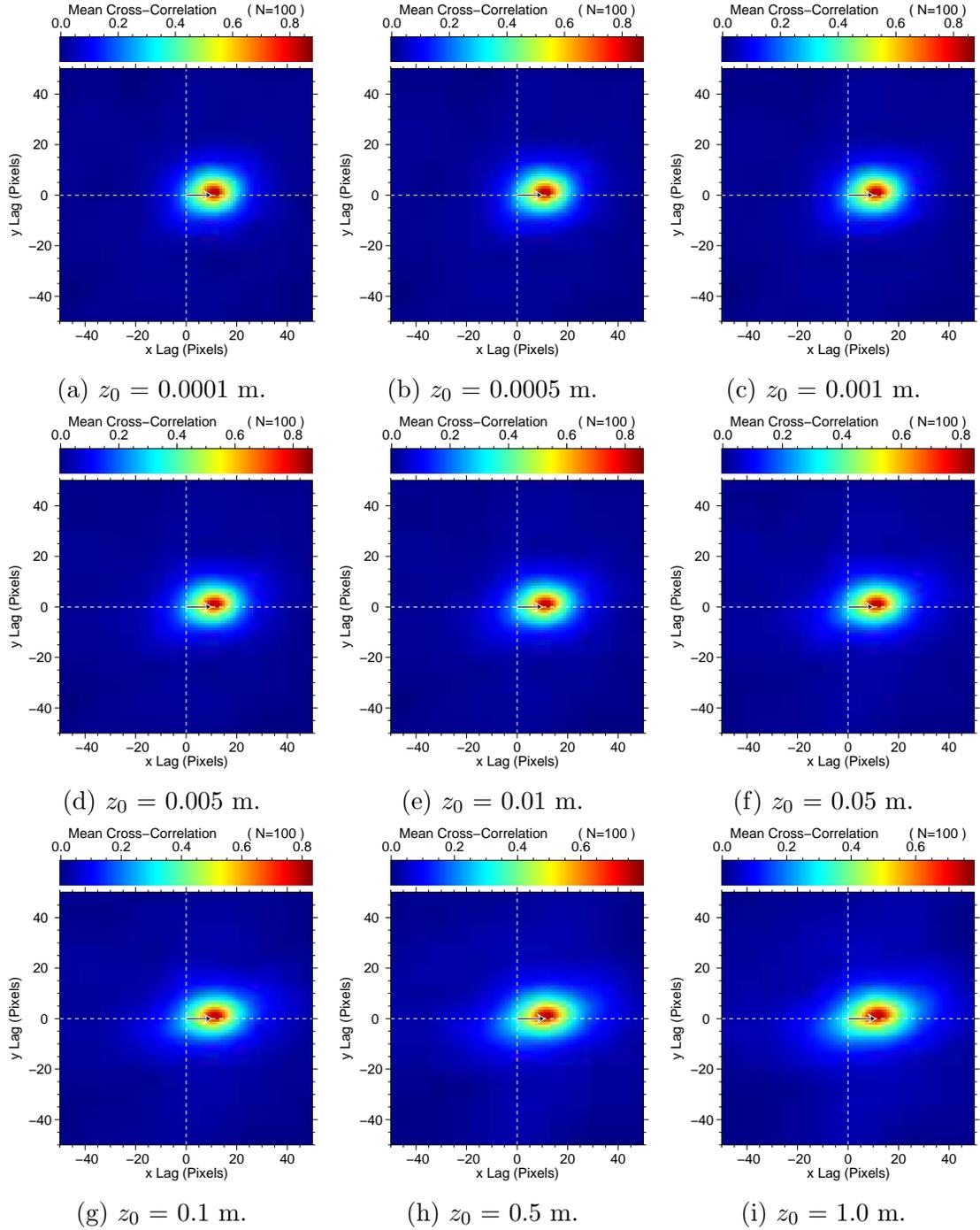


Figure 45. Mean cross-correlation functions for the velocity field  $(u, v) = (10 + u'(x, y), v'(x, y))$  (pixels/frames) with different values of the aerodynamic roughness length  $z_0$ .  $z_0$  changes from 0.0001 m to 1.0 m.

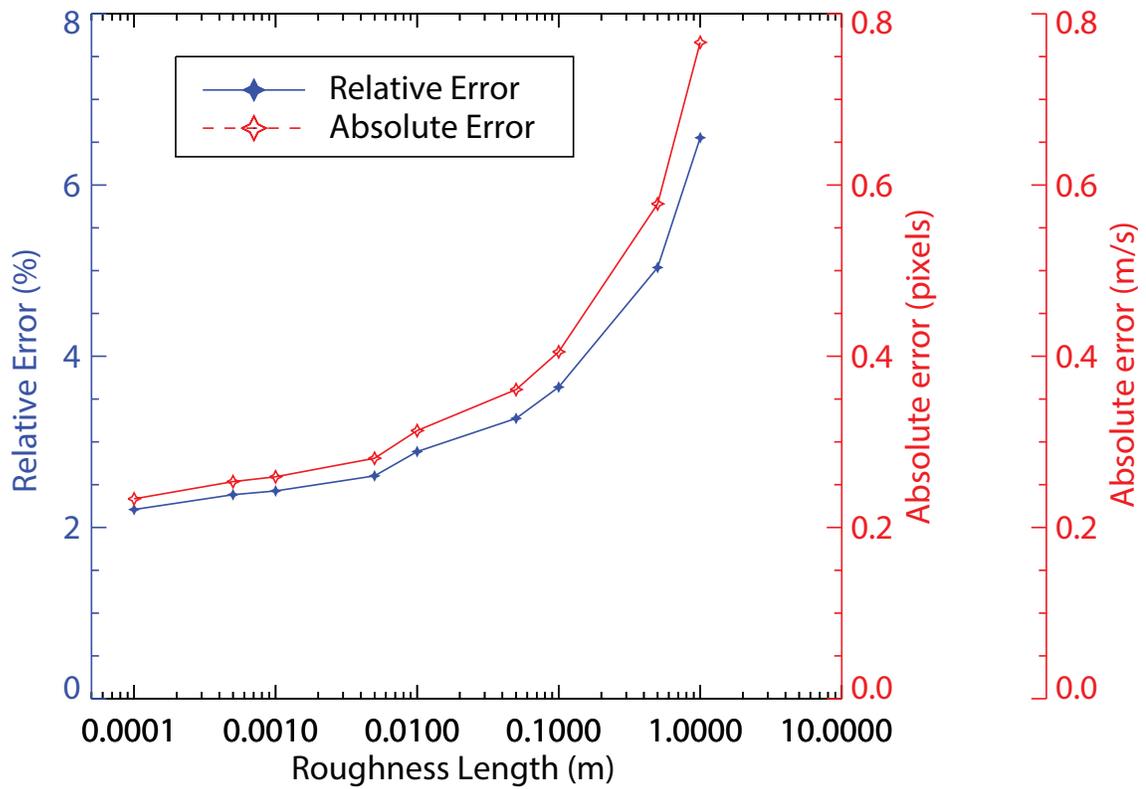


Figure 46. The relative error  $\left| \frac{D_{ccf} - D_{true}}{D_{true}} \right|$  vs the aerodynamic roughness length  $z_0$ , and the absolute error  $|D_{ccf} - D_{true}|$  vs the aerodynamic roughness length  $z_0$ .

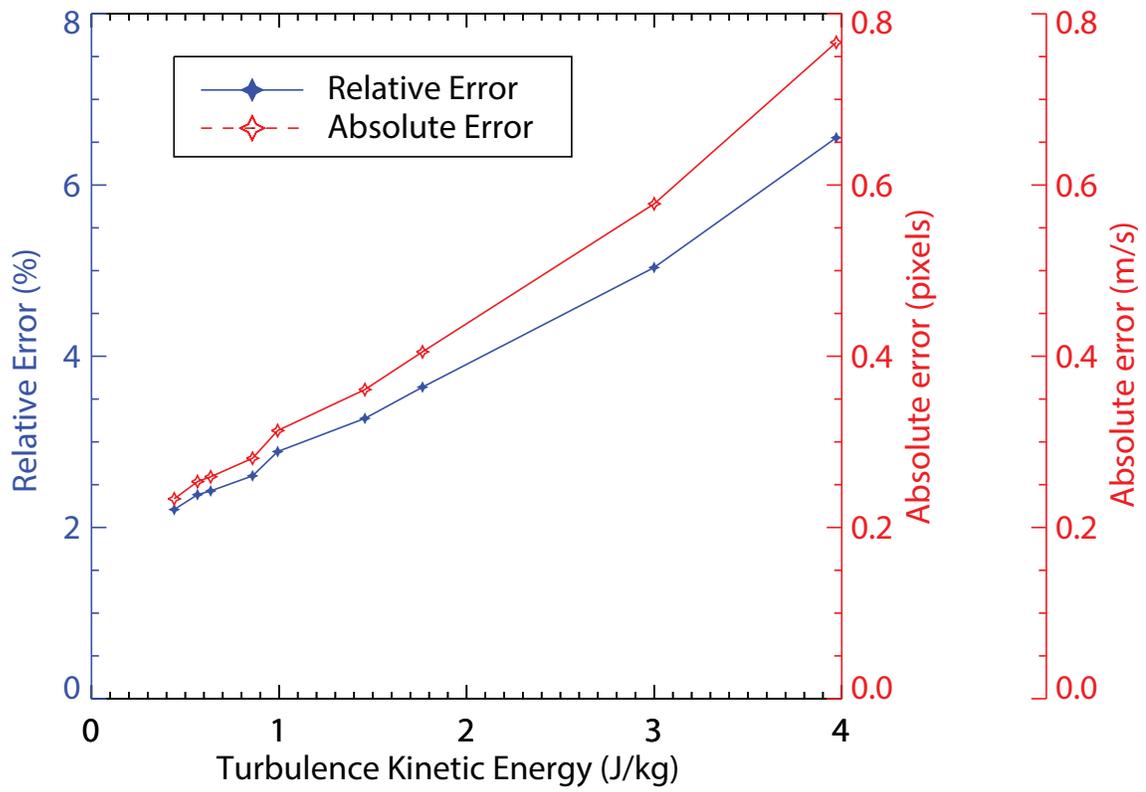


Figure 47. The relative error  $\left| \frac{D_{ccf} - D_{true}}{D_{true}} \right|$  versus the turbulence kinetic energy (TKE), and the absolute error  $|D_{ccf} - D_{true}|$  versus the turbulence kinetic energy (TKE).

### Effects of an Edge of a Scan Sector

The effects of a scan edge cutting a predominant aerosol feature was investigated by placing a Gaussian feature in the middle of a synthetic backscatter image and then cutting the feature by a diagonal edge line  $y = x$ . A uniform velocity field  $(u, v) = (10, 0)$  (pixels/ frames) is applied for  $N = 100$  pairs of synthetic backscatter images with a Gaussian feature cut by the edge line. Figure 48a represents the magnitude and direction of the velocity vectors at a given pixel location. Since every pixel location in the interrogation window has the same velocity vector, the mean and the standard deviation of the velocity vectors are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 0.00)$  (pixels/frames), respectively. Figure 48b shows the result of the cross-correlation algorithm for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(5.40, 4.46)$  (pixels) and  $(2.48, 2.43)$  (pixels), respectively. From the result, two important facts were observed. First, the resultant displacement vector is much lower than that predicted from a given velocity field. From the given velocity field, it was expected that the magnitude of the displacement would be  $D_{true} = 10.0$  pixels. However, the magnitude of the resultant displacement vector was  $D_{cef} = 7.00$  pixels which is 30% lower than  $D_{true}$ . In addition, the direction of the resultant displacement vector was different from that expected from the given velocity field. The given velocity field was in the x-direction. However, the resultant displacement vector was the angle  $39.6^\circ$  from the x-axis. This direction was close to the direction of the edge line (diagonal). The result is consistent with the observation (REAL lidar scan on June 24, 2013, California State University, Chico University Farm) as shown in Figure 32. Thus, the resultant displacement vector was inconsistent with the displacement expected from the given flow field.

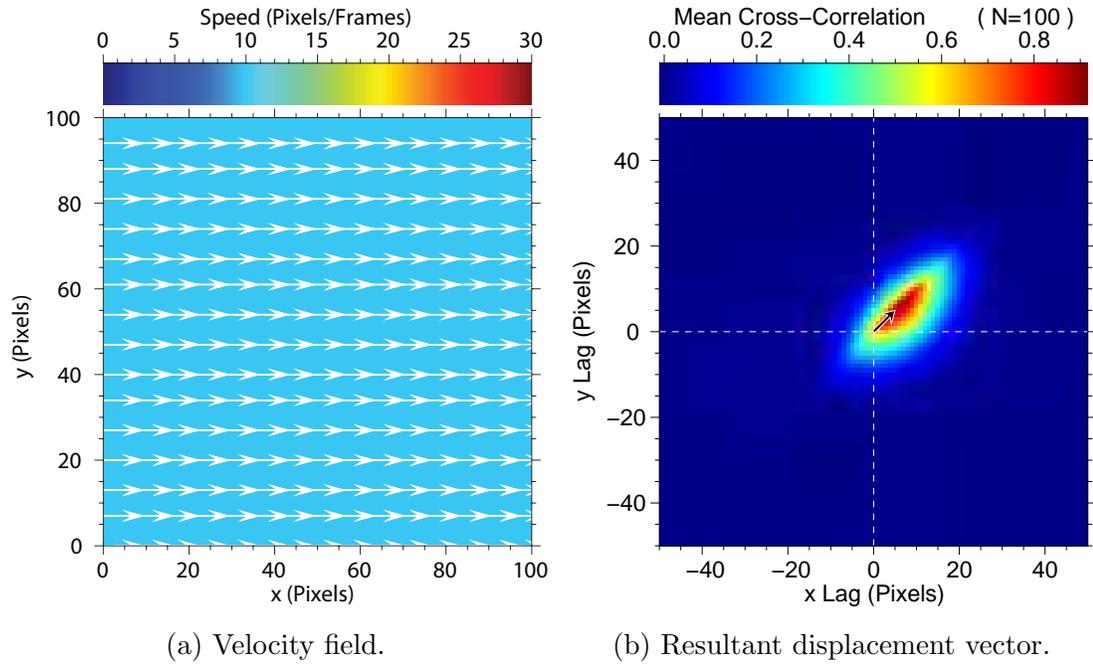


Figure 48. Panel a represents a non-turbulent uniform velocity field,  $(u, v) = (10, 0)$ , applied to block 1. The mean velocity and standard deviation are  $(10.0, 0.00)$  (pixels/frames) and  $(0.00, 0.00)$  (pixels/frames), respectively. Panel b is the mean cross-correlation function and resultant displacement vector for  $N = 100$  pairs of synthetic backscatter images. The mean and standard deviation of the displacement vector are  $(5.40, 4.46)$  (pixels) and  $(2.48, 2.43)$  (pixels), respectively.

## CHAPTER V

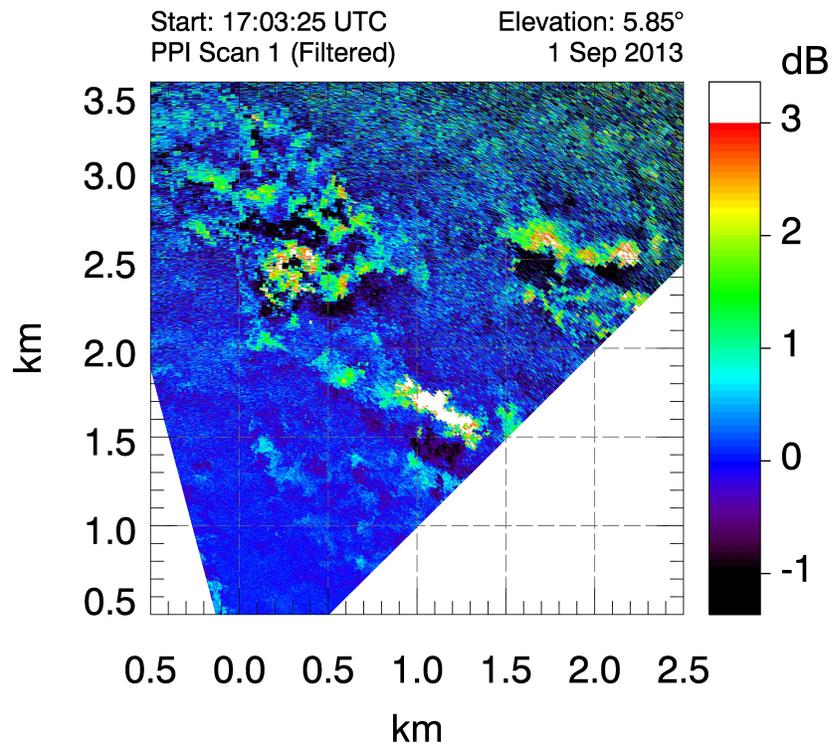
### DISCUSSION AND CONCLUSIONS

In this chapter, results from all tests of the cross-correlation algorithm using synthetic backscatter images and velocity fields are discussed. First, a synthetic backscatter image is compared with an actual lidar backscatter image. Next, validation of using the fast Fourier transform method is discussed. Then, the effect of zero-padding is examined by comparing the results using zero-padded images and non-zero-padded images. Next, the performance of the cross-correlation algorithm, associated with non-uniformity of the velocity field, is discussed by a series of examples, using two square pulses moving in the opposite directions. In addition, the edge effect is discussed by using a moving Gaussian feature cut by a diagonal edge line. Next, the performance of the cross-correlation algorithm, including under estimation of the velocity for non-uniform velocity fields, is summarized and discussed. Finally, conclusions are stated at the end of this chapter.

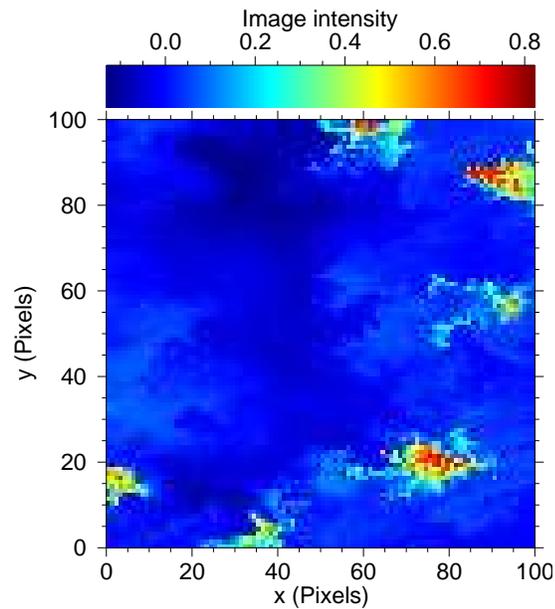
#### Synthetic Backscatter Images and Actual Lidar Data

Synthetic backscatter images are artificially created, and hence are not the same as the ones that are obtained from actual scans of the atmosphere. However, it is important to make synthetic backscatter images that are similar to the actual backscatter images. Figure 49a is an example of a backscatter image from a REAL lidar system (PPI scan). From Figure 49a, one can see that there are

several features in the region of approximately 1.0 – 3.0 km from the lidar system. Typical features in such an image are more than ten times brighter than that of the background. Figure 49b is an example of synthetic backscatter image used to test the cross-correlation algorithm. As discussed in previous sections, Gaussian features are placed randomly and diffused by turbulent perturbation field generated by the model described by Mann (1994) and Mann (1998). Comparing Figures 49a and 49b, one can see similarities and differences between them. The shape of actual aerosol features is strongly influenced by turbulent diffusion near the surface of the Earth, and synthetic backscatter images mimic the shape of such features. On the other hand, the actual aerosol backscatter images may contain coherent structures. For example, from Figure 49a, one can see an elongated aerosol feature across the scanned region. On the other hand, to make synthetic backscatter images Gaussian features are placed randomly in the interrogation window. In this case, a synthetic backscatter image does not contain any coherent structure. Thus, synthetic backscatter images may not represent the actual lidar backscatter image although both reflect turbulent diffusion of the atmosphere.



(a) An example of a REAL lidar backscatter image (PPI scan).



(b) An example of synthetic backscatter image.

Figure 49. Comparison of REAL lidar backscatter image and synthetic backscatter image

### Comparison of Covariance Method and FFT Method

The normalized cross-correlation function  $r_x$  for two waveforms  $f_1$  and  $f_2$  is defined as

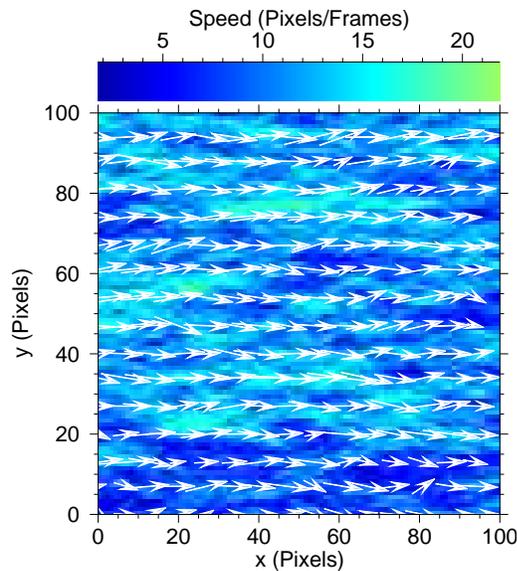
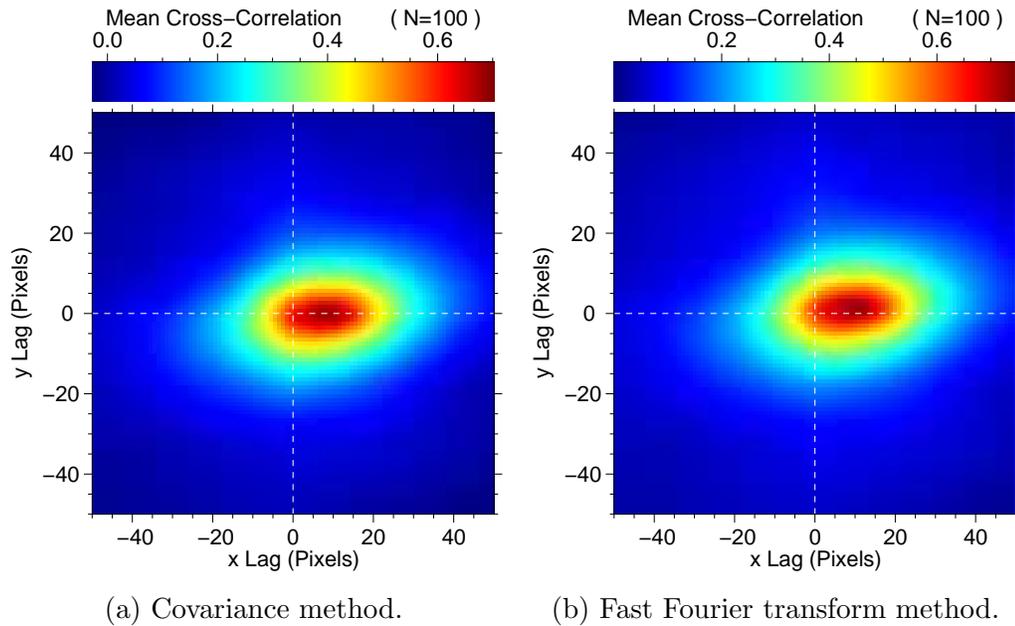
$$r_x = \frac{COV_{1,2}}{S_1 S_2}, \quad (1)$$

where  $COV_{1,2}$  is the covariance of the overlapped portions of  $f_1$  and  $f_2$ ,  $S_1$  is the standard deviation of  $f_1$ , and  $S_2$  is the standard deviation of  $f_2$ . One can use this function to estimate the wind velocity fields, but it is not very efficient. Instead, one can calculate the cross-correlation function  $r_x$  using the fast Fourier transform (FFT) method.

$$r_x = \frac{FFT^{-1}(FFT_1 FFT_2^*)}{S_1 S_2}, \quad (2)$$

where  $FFT_1 = \frac{\sum_{i=1}^N f_1 e^{-i2\pi kx}}{N}$  is the fast Fourier transform of  $f_1$ ,  $FFT_2 = \frac{\sum_{i=1}^N f_2 e^{-i2\pi kx}}{N}$  is the complex conjugate of the fast Fourier transform of  $f_2$ ,  $S_1$  is the standard deviation of  $f_1$ ,  $S_2$  is the standard deviation of  $f_2$ , and  $FFT^{-1}$  represents the inverse fast Fourier transform. This method (FFT method) is much more computationally efficient compared to the previous one (covariance method), so the FFT method was used to test the performance of the cross-correlation algorithm. To verify that the FFT method produces the same result as the covariance method, one applies a velocity field of strong turbulent perturbations to  $N = 100$  synthetic backscatter images, and calculates the mean cross-correlation function using the two methods. The velocity field was created by the sum of uniform flow and strong turbulent perturbations using the aerodynamic roughness length  $z_0 = 1.5$  m near the surface (altitude = 20 m). Figure 50c shows the velocity field applied to  $N = 100$  synthetic backscatter images. The mean velocity and standard deviation are (10.7, 0.26) (pixels/frames) and (3.00, 1.84) (pixels/frames),

respectively. The mean cross-correlation functions are shown in Figures 50a and 50b. Both methods gave the same mean displacement vector and standard deviation associated with different synthetic backscatter images. The mean and the standard deviation of the resultant displacement vectors, due to different synthetic backscatter images, are  $(9.31, 0.105)$  (pixels) and  $(1.53, 0.259)$  (pixels), respectively. However their peak values are 0.702 and 0.747, respectively, because of different normalization conditions. The results show that both methods give the same displacement vector, so it is reasonable to use the FFT method to calculate the cross-correlation function for computational efficiency.



(c) Velocity field.

Figure 50. The mean cross-correlation function for  $N = 100$  pairs of synthetic backscatter images is calculated by two different methods, the covariance method (panel a) and the fast Fourier transform method (panel b). Both panels a and b produces the same resultant displacement vector. However their peak values are 0.702 and 0.747, respectively, because of different normalization conditions. Panel c is the velocity field, with strong turbulent perturbations, applied to  $N = 100$  pairs of synthetic backscatter images.

### Effects of Zero-Padding

In the previous section, it was shown that the FFT method and the covariance method gave the same resultant displacement vector. Since, the FFT method is computationally more efficient than the covariance method, it is reasonable to apply the FFT method for real-time calculation of wind fields (Mauzey et al., 2012). However, according to Adrian and Westerweel (2011), the FFT method assumes periodic images, but synthetic images, like the actual atmosphere, are not periodic. Therefore, it is important to solve this problem by “zero-padding.” To confirm the intended result of zero-padding, one can perform the test of the cross-correlation algorithm by using a velocity field of strong turbulent perturbations. The velocity field was created by the sum of uniform flow  $(u, v) = (10, 0)$  (pixels/frames) and turbulent perturbation field produced by the model that is described by Mann (1994) and Mann (1998). Here, an aerodynamic roughness length  $z_0 = 2.0$  m and flow field altitude of 50 m were chosen. The velocity field was applied to  $N = 100$  pairs of synthetic backscatter images and the mean cross-correlation function and mean resultant displacement vector were calculated. To test the effects of zero-padding, one can compare the results of using zero-padded images and that of non zero-padded images, and all other factors being the same. The results are shown in Figure 51. Panels a and b are the mean cross-correlation functions with zero-padding and no zero-padding, respectively. Figure 51c represents the velocity field applied to  $N = 100$  pairs of synthetic backscatter images. The mean and the standard deviation of the velocity field are  $(11.8, 0.158)$  (pixels/frames) and  $(2.62, 1.62)$  (pixels/frames), respectively. The mean resultant vector and the standard deviation of the displacement vector, for zero-padded images, are  $(10.6, 0.0494)$  (pixels) and  $(1.86, 0.652)$  (pixels),

respectively. The percent difference of the magnitude of the resultant displacement vector and that predicted from the velocity field is 10.3%. On the other hand, corresponding results for non zero-padded images are (10.2, 0.0817) (pixels) and (2.30, 0.318) (pixels), respectively. The percent difference of the magnitude of the resultant displacement vector and that predicted from the velocity field is 13.2%. Comparing these results, one can see that the resultant displacement vector calculated by using zero-padded images is better than that of non zero-padded images. Thus, it is reasonable to use zero-padded images for applying the cross-correlation algorithm to real lidar data.

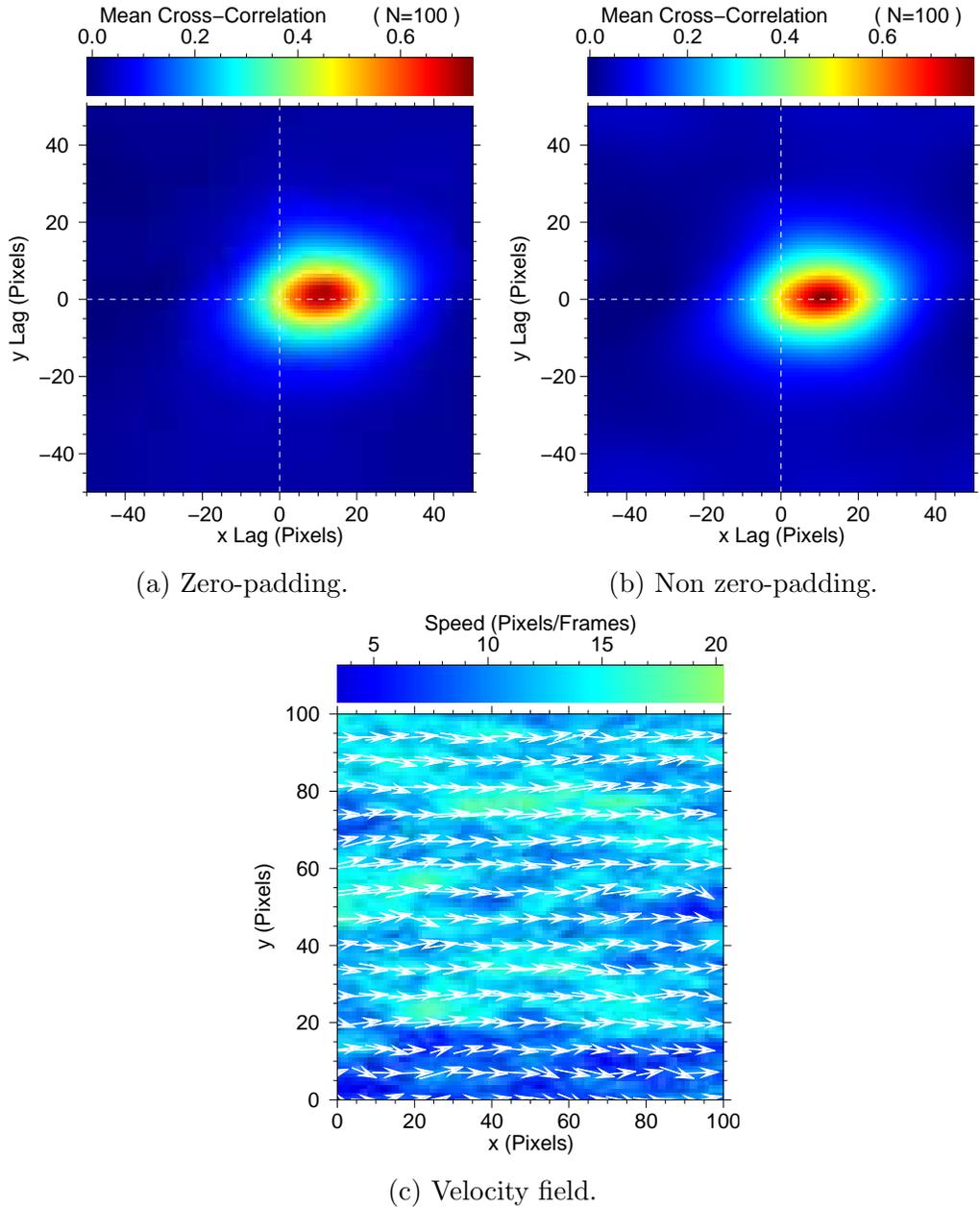


Figure 51. The mean cross-correlation function for  $N = 100$  pairs of synthetic backscatter images is calculated by two different methods, zero-padding FFT method and non-zero-padding FFT method. Panels a and b show the mean cross-correlation functions of zero-padding and non-zero-padding methods, respectively. Although they are similar, zero-padding method produces the displacement vector closer to that predicted from the velocity field as shown in panel c.

## Effects of Non-Uniformity

From analytical studies of a 2-D rectangular pulse moving at a constant velocity, the cross-correlation function  $r_{x_{lag}, y_{lag}}$  calculated from the images  $f_1(x, y)$  and  $f_2(x, y)$ , can be expressed as

$$r_{x_{lag}, y_{lag}} = \begin{cases} \left( \frac{I_0^2}{S_1 S_2} \right) [a + (x_{lag} - 10)][b + (y_{lag} - 10)] & : 10 - a \leq x_{lag} \leq 10 : 10 - b \leq y_{lag} \leq 10 \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a - (x_{lag} - 10)][b - (y_{lag} - 10)] & : 10 \leq x_{lag} \leq 10 + a : 10 \leq y_{lag} \leq 10 + b \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a + (x_{lag} - 10)][b - (y_{lag} - 10)] & : 10 - a \leq x_{lag} \leq 10 : 10 \leq y_{lag} \leq 10 + b \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a - (x_{lag} - 10)][b + (y_{lag} - 10)] & : 10 \leq x_{lag} \leq 10 + a : 10 - b \leq y_{lag} \leq 10 \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the given Gaussian feature,  $a$  is the width of the rectangular pulse,  $b$  is the height of the rectangular pulse,  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ . From this equation, one can see that the peak value and profile of the cross-correlation function depend on the peak intensity  $I_0$  and the area  $A = ab$  of the rectangular pulse. That is, the cross-correlation function is likely biased if a bright (high intensity) and large (large area) feature appears in both  $f_1(x, y)$  and  $f_2(x, y)$ . In addition, one may expect that the effect of the peak intensity is greater than that of the area since the cross-correlation function is proportional to the peak intensity squared. The bias of the cross-correlation algorithm is easy to observe by generating two square pulses and applying different velocities. Here, some examples of two square pulses (features 1 and 2) moving at different velocities in the time interval  $\Delta t = t_2 - t_1$  are shown, which demonstrates that the resultant displacement vectors are biased. Table 4 shows a series of examples of the performance of the cross-correlation algorithm in the case of two square features moving at different velocities. Feature 1 has the peak intensity  $I_0 = 1.0$ , the area

$A = (10 \times 10)$  pixels, and moves 10 units in the positive x-direction in the time interval  $\Delta t = t_2 - t_1$ . On the other hand, feature 2 moves 10 units in the negative x-direction and its peak intensity and area are varied.

TABLE 4  
LIST OF NON-UNIFORMITY TESTS OF THE  
CROSS-CORRELATION ALGORITHM

Test	Feature 1, $(u, v) = (10, 0)$	Feature 2, $(u, v) = (-10, 0)$
1	$I_0 = 1.0, A = (10 \times 10)pixels$	$I_0 = 1.0, A = (4 \times 4)pixels$
2	$I_0 = 1.0, A = (10 \times 10)pixels$	$I_0 = 1.0, A = (20 \times 20)pixels$
3	$I_0 = 1.0, A = (10 \times 10)pixels$	$I_0 = 0.5, A = (10 \times 10)pixels$
4	$I_0 = 1.0, A = (10 \times 10)pixels$	$I_0 = 2.0, A = (10 \times 10)pixels$
5	$I_0 = 1.0, A = (10 \times 10)pixels$	$I_0 = 0.5, A = (14 \times 14)pixels$
6	$I_0 = 1.0, A = (10 \times 10)pixels$	$I_0 = 0.5, A = (20 \times 20)pixels$
7	$I_0 = 1.0, A = (10 \times 10)pixels$	$I_0 = 0.5, A = (24 \times 24)pixels$

Tests of non-uniformity in the case of two features (Features 1 and 2) moving at different velocities. Feature 1 and 2 move 10 units in the positive and the negative x-direction, respectively, in the time interval  $\Delta t = t_2 - t_1$ .

#### Effect of the Area of Features

Figure 52 shows the case that features 1 and 2 have the same peak intensity but feature 2 is smaller than feature 1. Panels a and b represent the images at  $t_1$  and  $t_2$ , respectively. The cross-correlation result is shown in panel c. Figure 52c reveals that there are four peaks and two of them represent the displacements of features 1 and 2. Here, the cross-correlation algorithm selects the displacement of feature 1 since the peak associated with the motion of feature 1 is larger than that of feature 2. The result shows that the cross-correlation algorithm selects the displacement of the feature with larger area.

Figure 53 shows the case that features 1 and 2 have the same peak intensity but feature 2 is larger than feature 1. Panels a and b represent the images at  $t_1$  and  $t_2$ , respectively. The cross-correlation result is shown as Figure 53c. From Figure 53c one can see that there are four peaks (two of them are overlapped) and two of them represent the displacements of features 1 and 2. Here, the cross-correlation algorithm selects the displacement of feature 2 since the peak associated with the motion of feature 2 is brighter than that of feature 1. As in the previous result, the cross-correlation algorithm is biased and selects the displacement of the feature with larger area.

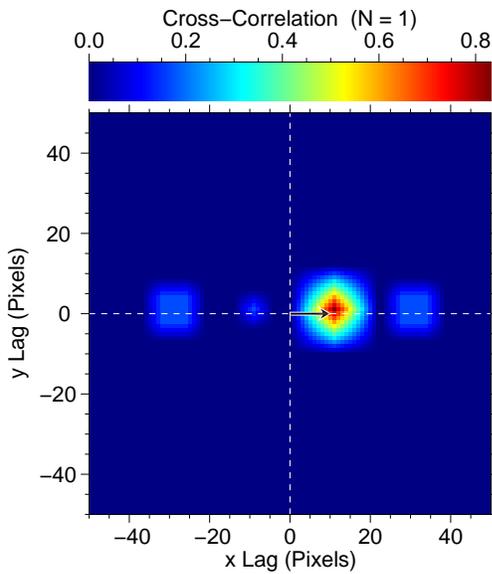
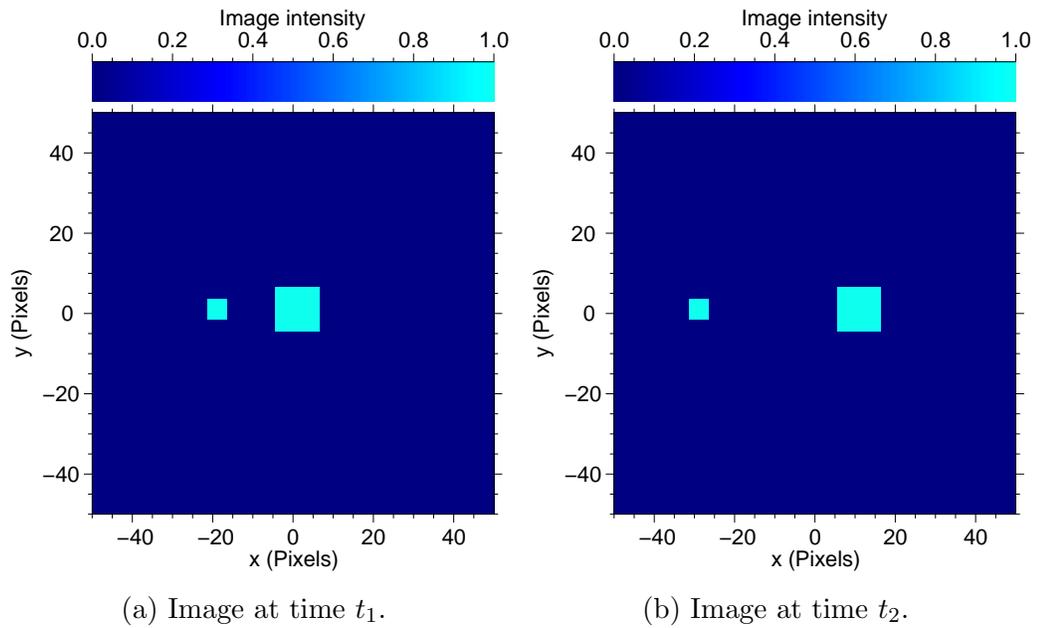


Figure 52. Panels a and b show that two square features 1 and 2 move 10 units in opposite directions in the time interval  $\Delta t = t_2 - t_1$ . Feature 1 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the positive  $x$ -direction. Feature 2 has the dimension  $4 \times 4$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the negative positive  $x$ -direction. Panel c shows the cross-correlation function and resultant displacement vector applied to these images.

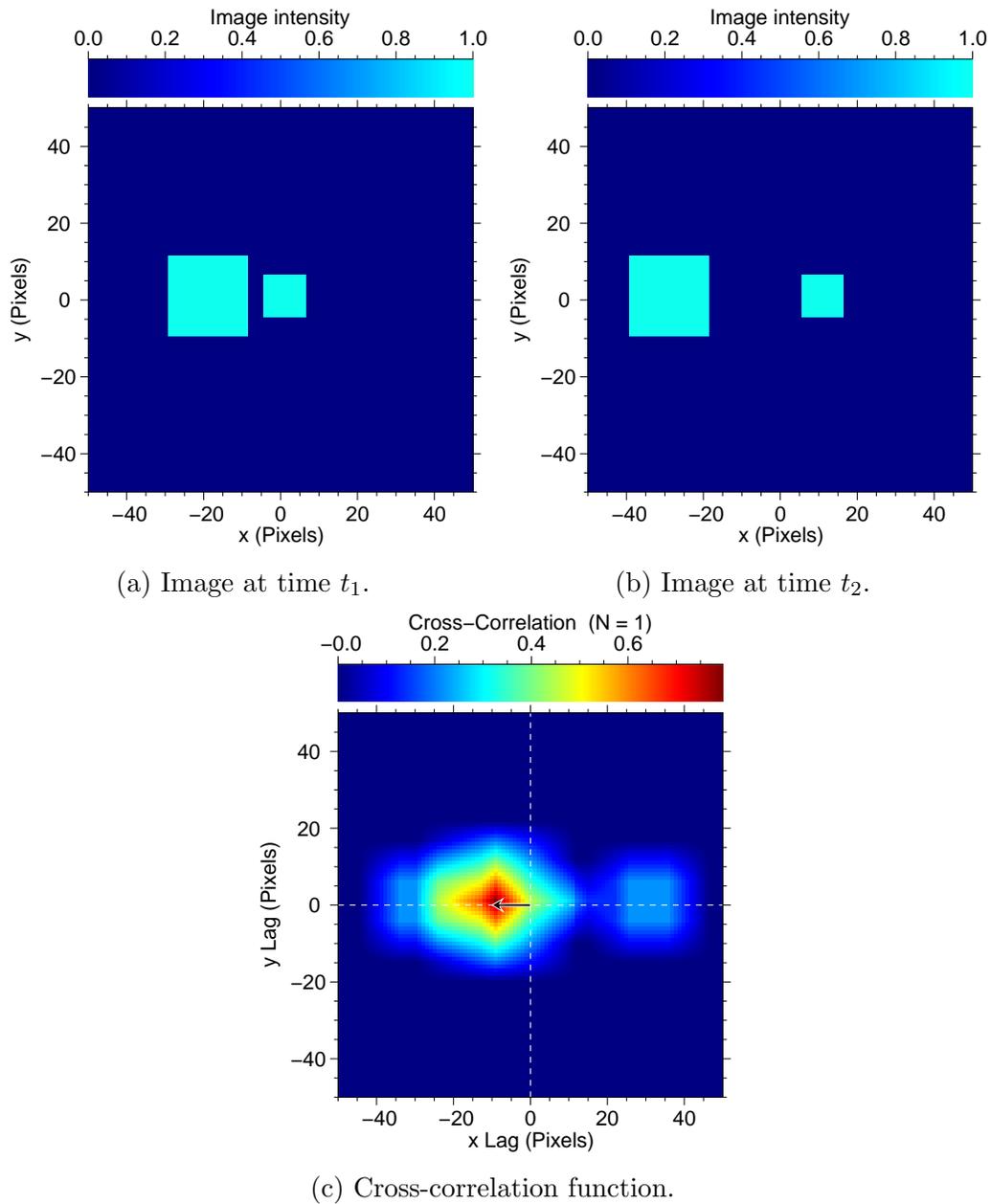


Figure 53. Panels a and b show that two square features 1 and 2 move 10 units in opposite directions in the time interval  $\Delta t = t_2 - t_1$ . Feature 1 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the positive x-direction. Feature 2 has the dimension  $20 \times 20$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the negative positive x-direction. Panel c shows the cross-correlation function and resultant displacement vector applied to these images.

### Effect of the Intensity of Features

Figure 54 shows the case that features 1 and 2 have the same area but feature 2 has lower peak intensity than that of feature 1. Panels a and b represent the images at  $t_1$  and  $t_2$ , respectively. The cross-correlation result is shown in panel c. From Figure 54c one can see that there are four peaks and two of them represent the displacements of features 1 and 2. Here, the cross-correlation algorithm selects the displacement of feature 1 since the peak associated with the motion of feature 1 is larger than that of feature 2. The result shows that the cross-correlation algorithm is biased and selects the displacement of the brighter feature.

Figure 55 shows the case where features 1 and 2 have the same area but feature 2 has the greater peak intensity than that of feature 1. Panels a and b represent the images at  $t_1$  and  $t_2$ , respectively. The cross-correlation function result is shown in panel c. From it one can see that there are four peaks and two of them represent the displacements of features 1 and 2. Here, the cross-correlation algorithm selects the displacement of feature 2 since the peak associated with the motion of feature 2 is larger than that of feature 1. As in the previous result, the cross-correlation algorithm selects the displacement of the brighter feature.

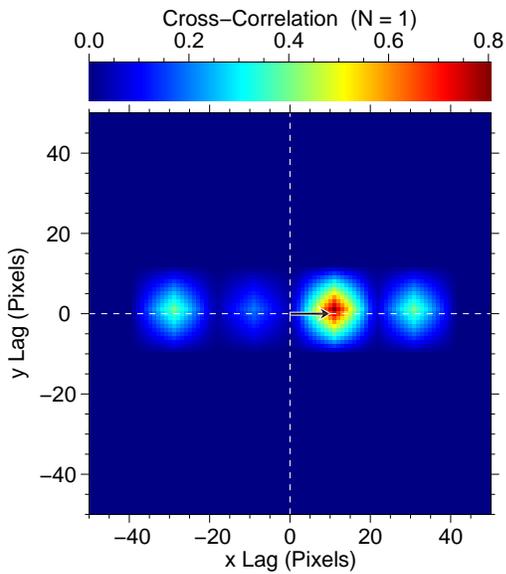
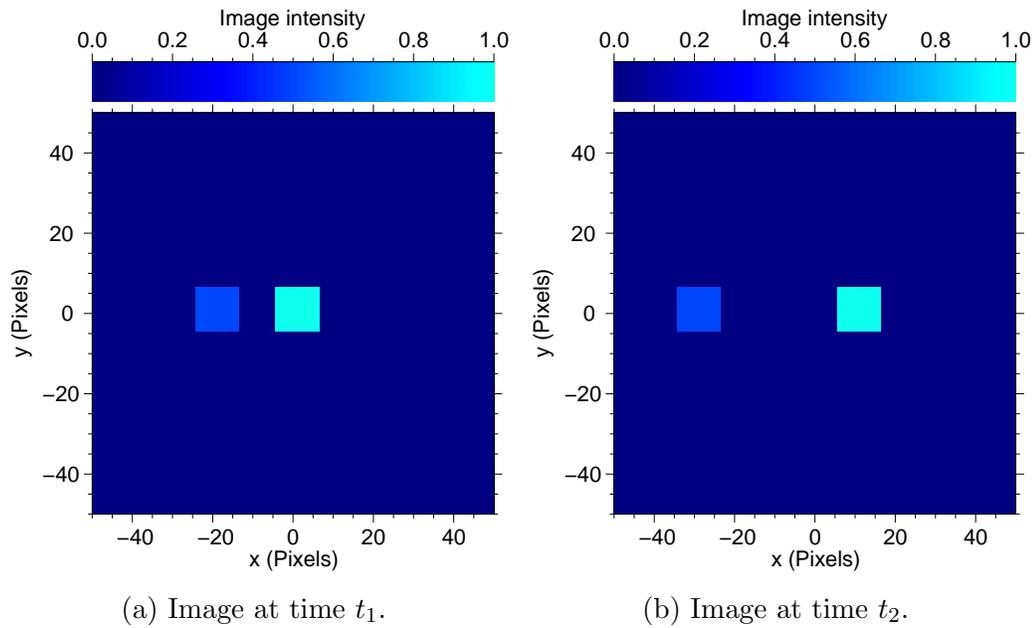


Figure 54. Panels a and b show that two square features 1 and 2 move 10 units in opposite directions in the time interval  $\Delta t = t_2 - t_1$ . Feature 1 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the positive x-direction. Feature 2 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 0.5$ , and moves 10 units in the negative positive x-direction. Panel c shows the cross-correlation function and resultant displacement vector applied to these images.

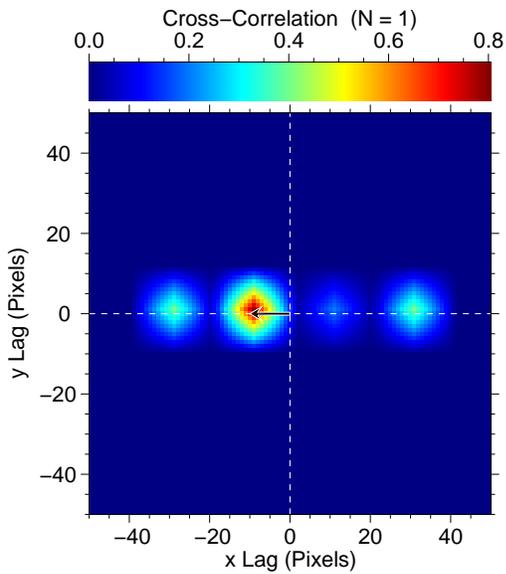
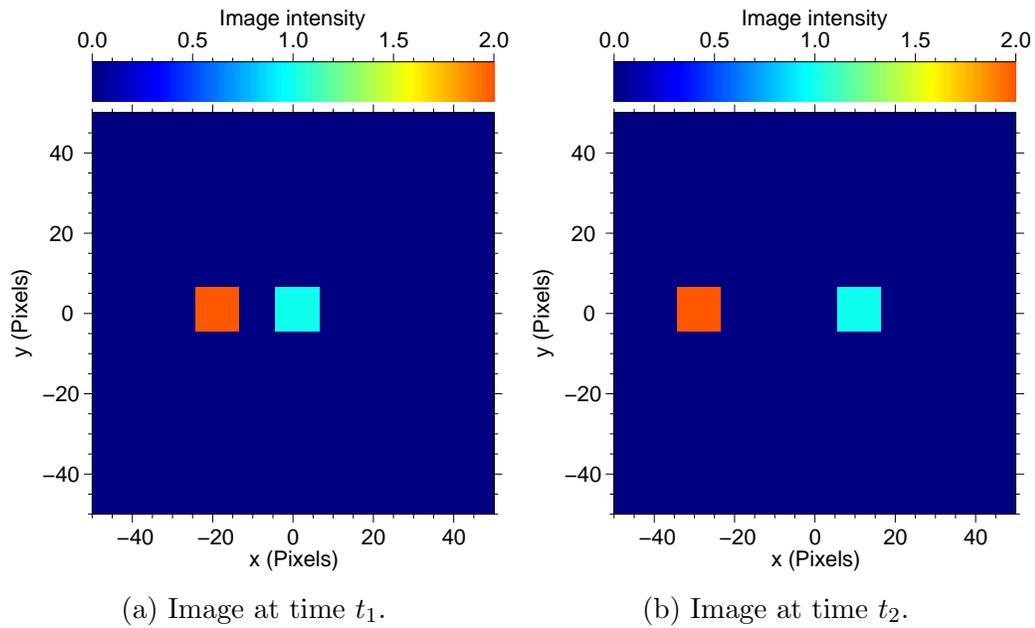


Figure 55. Panels a and b show that two square features 1 and 2 move 10 units in opposite directions in the time interval  $\Delta t = t_2 - t_1$ . Feature 1 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the positive x-direction. Feature 2 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 2.0$ , and moves 10 units in the negative positive x-direction. Panel c shows the cross-correlation function and resultant displacement vector applied to these images.

Effect of Both Area and  
Intensity of Features

Figure 56 shows the case that feature 2 is larger in area ( by approximately  $2\times$ ) than feature 1, but its intensity is lower (by approximately  $0.5\times$ ) than that of feature 1. Panels a and b represent the images at  $t_1$  and  $t_2$ , respectively. The cross-correlation result is shown in panel c. From Figure 56c one can see that there are four peaks and two of them represent the displacements of features 1 and 2. Here, the cross-correlation algorithm selects the displacement of feature 1 since the peak associated with the motion of feature 1 is larger than that of feature 2. From the result, one can see that the effect of the intensity of features is greater than that of their areas. The cross-correlation algorithm selects the displacement of the brighter feature although both the intensity and the area affect the shape of the cross-correlation function.

Figure 57 shows the case that feature 2 is larger in are (by approximately  $4\times$ ) than feature 1, but its intensity is lower (by approximately  $0.5\times$ ) than that of feature 1. Panels a and b represent the images at  $t_1$  and  $t_2$ , respectively. The cross-correlation result is shown in panel c. From Figure 57c one can see that four peaks exist and two of them represent the displacements of features 1 and 2. Here, the cross-correlation algorithm selects the displacement of feature 1 since the peak associated with the motion of feature 1 is larger than that of feature 2. As in the previous case, one can see that the effect of the intensity of features is greater than that of areas. The cross-correlation algorithm selects the displacement of the brighter feature although both the intensity and the area affect the resultant displacement vector.

Figure 58 shows the case that feature 2 is larger in area (more than  $4\times$ ) than feature 1, but its peak intensity is lower (by approximately  $0.5\times$ ) than that of

feature 1. Panels a and b represent the images at  $t_1$  and  $t_2$ , respectively. The cross-correlation result is shown as Figure 58c. From Figure 58c one can see that there are four peaks and two of them represent the displacements of features 1 and 2. Here, the cross-correlation algorithm selects the displacement of feature 2 since the peak associated with the motion of feature 2 is brighter than that of feature 1. The result shows that the cross-correlation algorithm could select the displacement of the larger and dimmer feature, although the effect of intensity of features is much greater than that of their areas.

From the results of a series of non-uniformity tests, the following observations were made:

- (1) the cross-correlation algorithm selects the displacement of a larger feature if the intensities of features are the same,
- (2) the cross-correlation algorithm selects the displacement of a brighter feature if the areas of features are the same, and
- (3) the intensity and the area are both influencing factors of the cross-correlation algorithm, but the effect of the intensity is much greater than that of the area (more than  $4\times$ ).

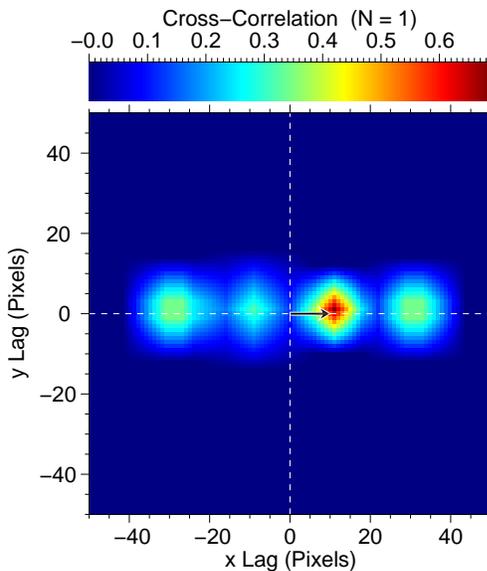
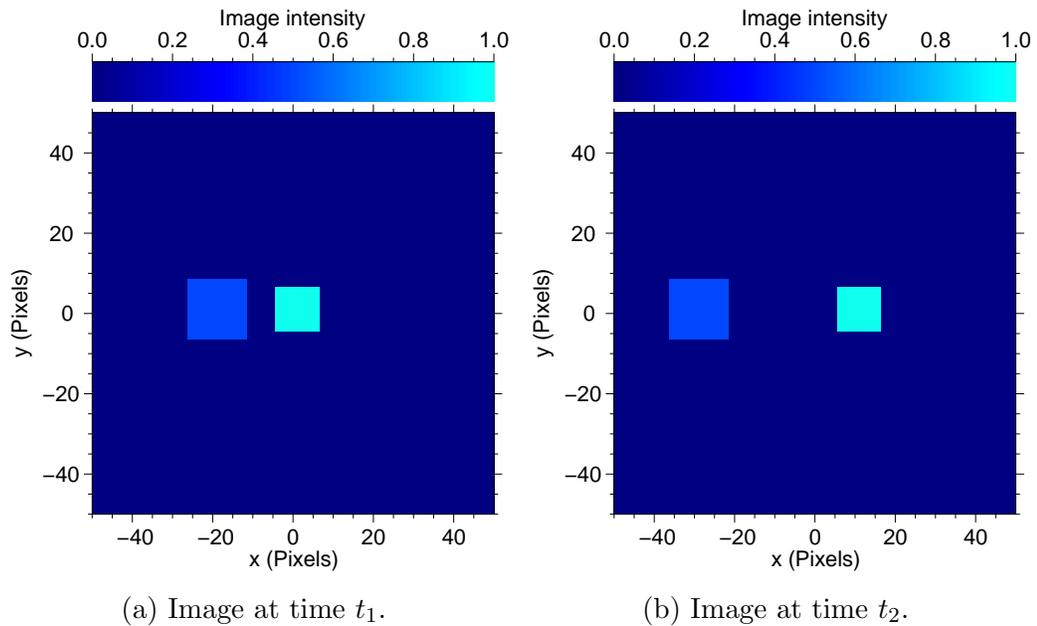


Figure 56. Panels a and b show that two square features 1 and 2 move 10 units in opposite directions in the time interval  $\Delta t = t_2 - t_1$ . Feature 1 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the positive x-direction. Feature 2 has the dimension  $14 \times 14$  pixels, the peak intensity  $I_0 = 0.5$ , and moves 10 units in the negative positive x-direction. Panel c shows the cross-correlation function and resultant displacement vector applied to these images.

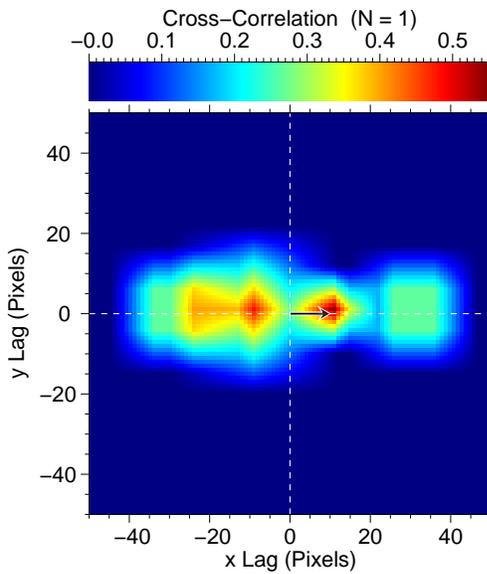
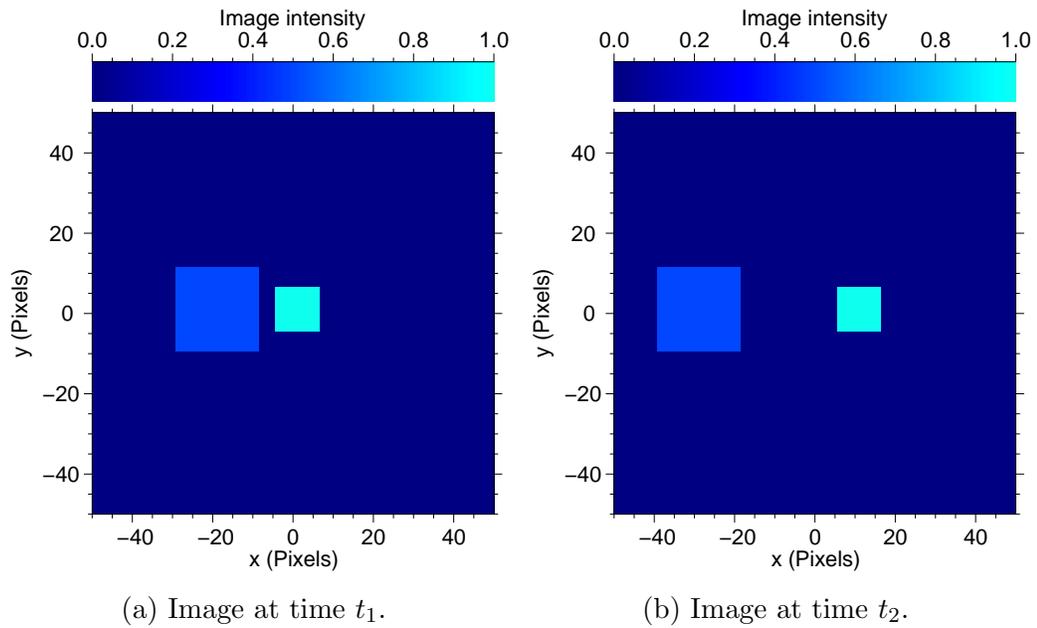


Figure 57. Panels a and b show that two square features 1 and 2 move 10 units in opposite directions in the time interval  $\Delta t = t_2 - t_1$ . Feature 1 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the positive x-direction. Feature 2 has the dimension  $20 \times 20$  pixels, the peak intensity  $I_0 = 0.5$ , and moves 10 units in the negative positive x-direction. Panel c shows the cross-correlation function and resultant displacement vector applied to these images.

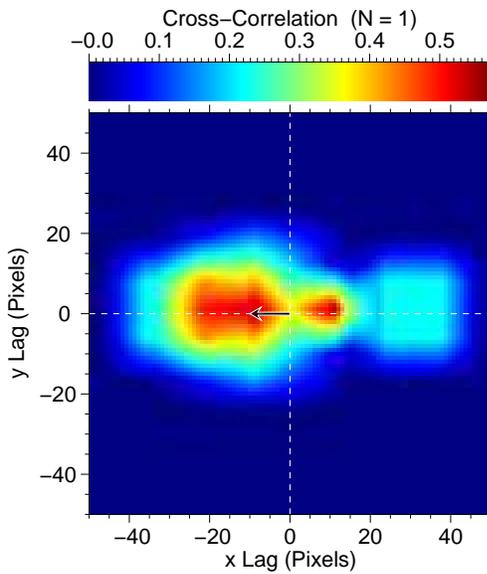
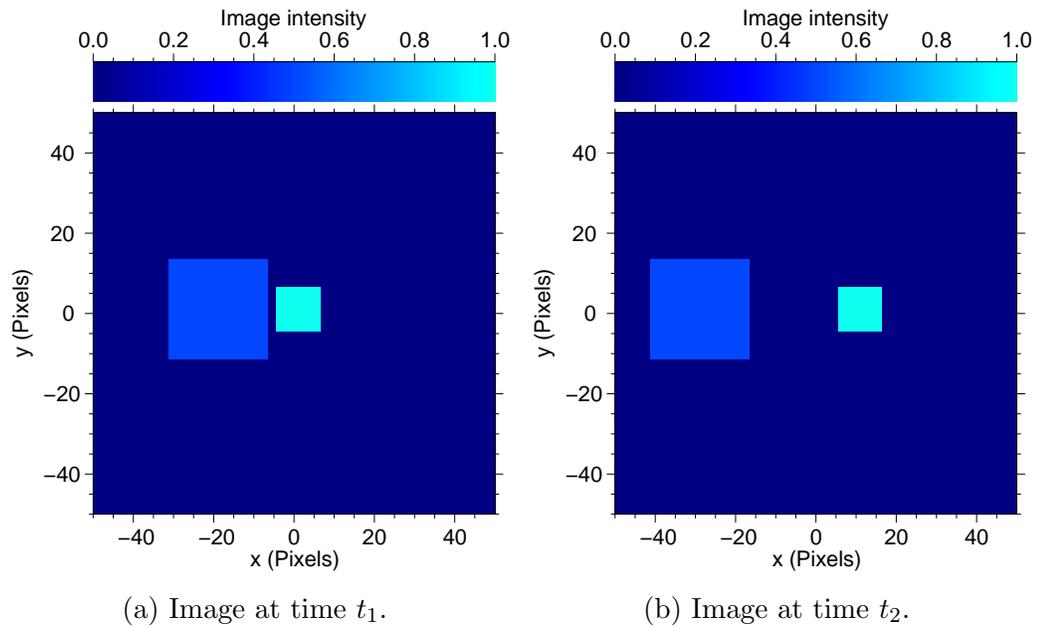
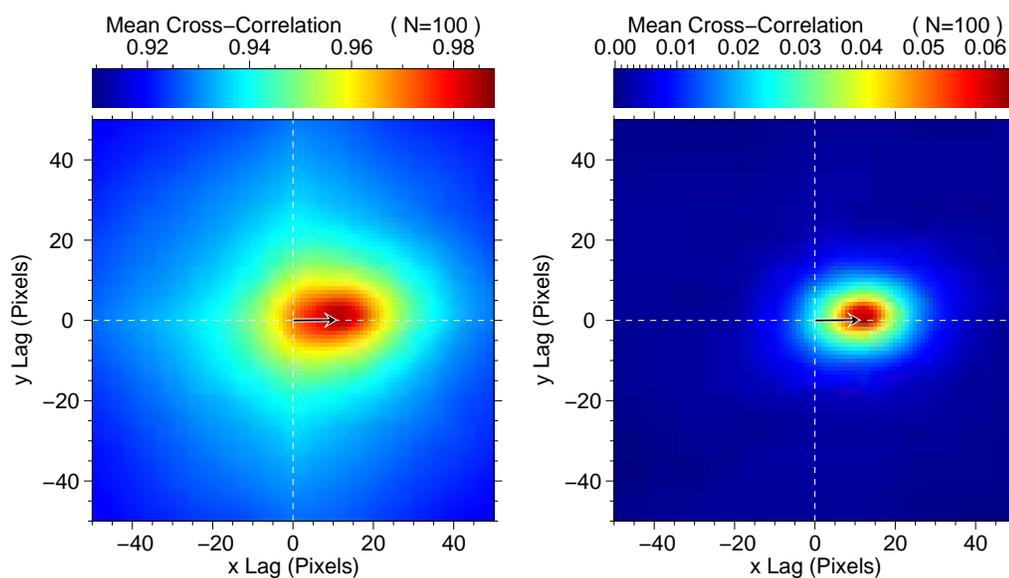


Figure 58. Panels a and b show that two square features 1 and 2 move 10 units in opposite directions in the time interval  $\Delta t = t_2 - t_1$ . Feature 1 has the dimension  $10 \times 10$  pixels, the peak intensity  $I_0 = 1.0$ , and moves 10 units in the positive x-direction. Feature 2 has the dimension  $24 \times 24$  pixels, the peak intensity  $I_0 = 0.5$ , and moves 10 units in the negative positive x-direction. Panel c shows the cross-correlation function and resultant displacement vector applied to these images.

### Effects of Histogram Equalization

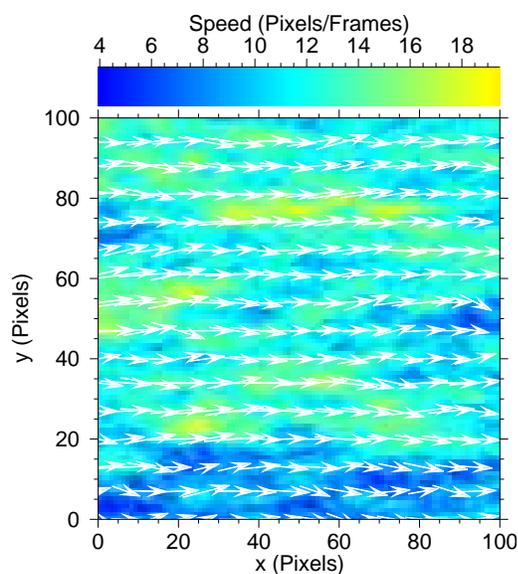
From the previous section, the results of non-uniformity tests show that the intensity of features influences the performance of the cross-correlation algorithm. That is, the cross-correlation algorithm is likely biased by the motion of bright features in the interrogation window. For example, the cross-correlation algorithm may underestimate the wind vector if a large and bright feature moves at lower speed than other dim and small features in the interrogation window. In this case, one may apply histogram equalization to images before computing the cross-correlation function. Histogram equalization is a method in image processing that allows the brightness of features to be adjusted by using the image's histogram. Schols and Eloranta (1992) used histogram equalization to distribute the influence of aerosol features and compute the horizontal wind vectors from lidar backscatter images. To test the effects of histogram equalization, mean displacement vectors were computed with and without histogram equalization under controlled conditions. Here, the velocity field of relatively strong turbulent flow (the aerodynamic roughness length  $z_0 = 1.5$  m; flow field altitude of 50 m) is applied to  $N = 100$  pairs of synthetic backscatter images with and without histogram equalization, and the mean resultant displacement vectors and the standard deviation associated with different synthetic backscatter images are computed. Figure 59 shows the results of the tests for histogram equalization. Panels a and b show the mean cross-correlation functions with and without histogram equalization, respectively. Figure 59c shows the velocity field used to generate the second image of  $N = 100$  pairs of synthetic backscatter images. The mean and the standard deviation of the velocity field are (11.6, 0.145) (pixels/frames) and (2.40, 1.48) (pixels/frames), respectively. The results show that

the mean and the standard deviation of the resultant displacement vectors, with histogram equalization are  $(10.9, 0.0860)$  (pixels) and  $(0.449, 0.105)$  (pixels), respectively. That is, the magnitude of the resultant vector is 6.31% lower than that expected from the given velocity field. On the other hand, the mean and the standard deviation of the resultant displacement vectors, without histogram equalization are  $(11.2, 0.0899)$  (pixels) and  $(1.05, 0.409)$  (pixels), respectively. That is, the magnitude of the resultant vector is 3.36% lower than that expected from the given velocity field. Thus, the performance of the cross-correlation algorithm, for this synthetic test, is likely decreased by applying histogram equalization. One of the major impacts of histogram equalization is to lower the performance of the cross-correlation algorithm by broadening the peak increasing ambiguity of the peak location. Comparing Figures 59a and 59b, one can see that the peak of the mean cross-correlation with histogram equalization is broader compared to the one without histogram equalization. In this case, it is more difficult to locate the peak via polynomial fitting, and such difficulty affects the performance of the cross-correlation algorithm.



(a) Histogram equalization.

(b) Non histogram equalization.



(c) Velocity field.

Figure 59. The mean cross-correlation function for  $N = 100$  pairs of synthetic backscatter images is calculated by two different methods, histogram equalization of synthetic backscatter images applied before computing the cross-correlation function, and without histogram equalization. Panels a and b show the mean cross-correlation functions with histogram equalization and non histogram equalization, respectively. The results show that non histogram equalization produces the displacement vector, closer to that predicted from the velocity field as shown in panel c.

### Effects of the Edge of a Scan Sector

When one scans the atmosphere horizontally with an aerosol lidar (PPI scan), the shape of the image is a sector. In this case, one can extract full square blocks in the middle of the scan area, but one cannot obtain full blocks if they extend beyond the edge of the scan. If a predominant feature in the scan area is cut by an edge line, it is difficult to estimate the wind velocity. One example of this problem is demonstrated in Figure 60. It shows a Gaussian feature of the peak intensity  $I_0 = 1.0$  cut by a diagonal edge line  $y = x$ . The image of a Gaussian feature that moves 10 units in the positive x-direction in the time interval  $\Delta t = t_2 - t_1$  is shown at time  $t_2$  in Figure 60b. Here, one can see that more than half of the Gaussian feature disappears due to its motion. Figure 60c shows the cross-correlation function and the resultant displacement vector corresponding to these images. The resultant displacement vector is (5.0, 5.0) (pixels) which is different from the mean of the true velocity field (10, 0) (pixels). From this example, the following observations associated with the effects of an edge were made:

- (1) the magnitude of the resultant displacement vector is lower than that expected from the velocity field if a predominant feature is cut by an edge line, and
- (2) the direction of the resultant displacement vector tends to be parallel to an edge line if a predominant feature is cut by an edge line.

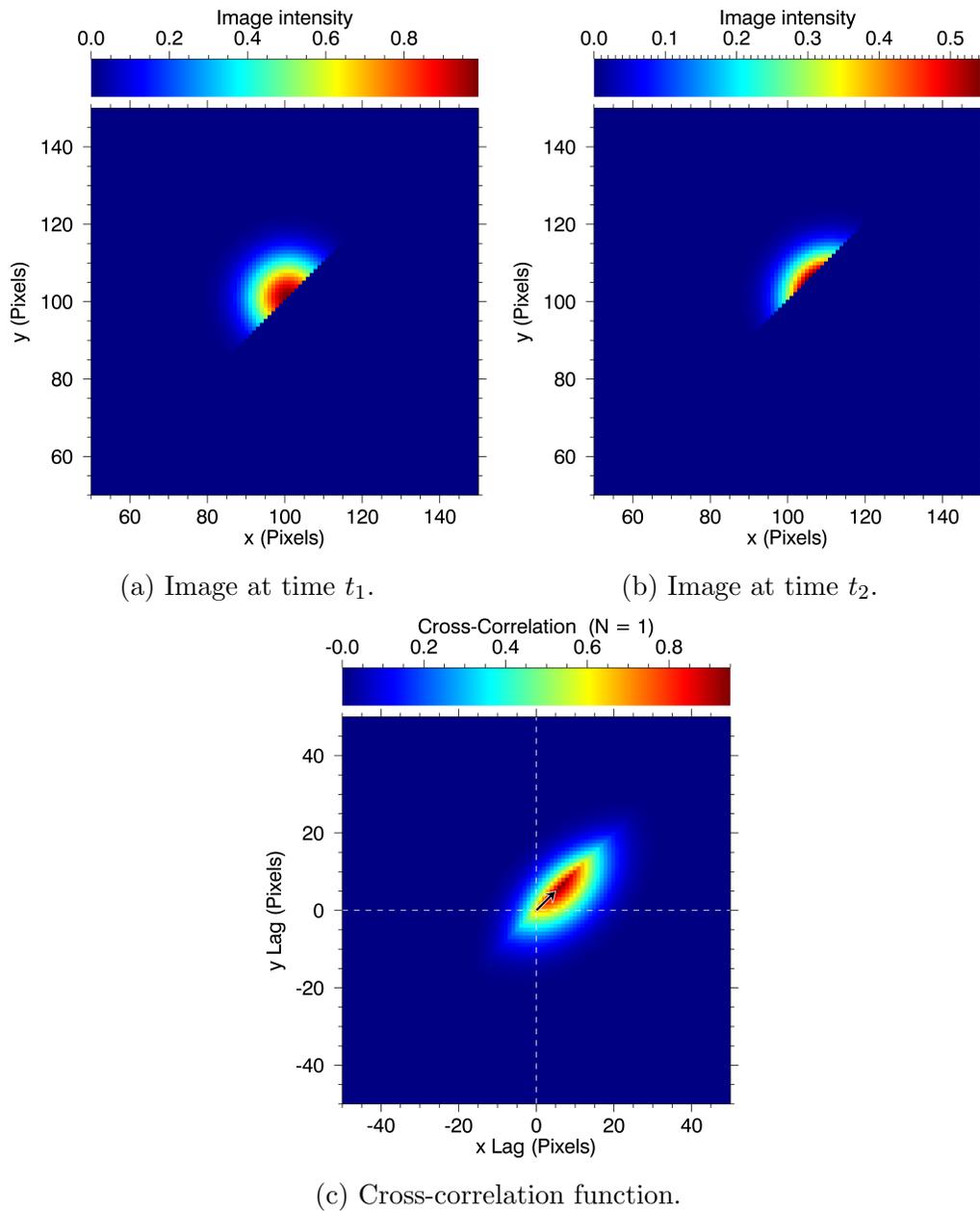


Figure 60. Effects of image edge are investigated by using a Gaussian feature cut by an edge line. Panels a and b show that the Gaussian feature moves 10 units in the positive x-direction in the time interval  $\Delta t = t_2 - t_1$ , and is cut by an edge line  $y = x$ . Panel c shows the cross-correlation function and the resultant displacement vector applied to these images. The resultant displacement vector is (5.00, 5.00) pixels which is different from the expected displacement vector (10.0, 0.00) pixels.

## Features Cut by the Edge of the Interrogation Window

### Description of the Problem

When aerosol features move in the time interval  $\Delta t = t_2 - t_1$  between consecutive scans, some features may straddle the edges of the interrogation window. In this case, such features are cut by the edges of the block. To investigate the effects of straddling features, a rectangular feature of dimensions  $20 \times 10$  pixels with a constant velocity  $(u, v) = (10.0, 0.00)$  (pixels/frames) is used. Figure 61 shows one example of the rectangular feature entering the interrogation window. Panel a shows the feature straddling the interrogation window at  $t_1$ . The feature is cut by a block edge. Panel b shows the feature at  $t_2$ . After moving 10 units in the positive x-direction, the feature is entirely within the interrogation window. Panel c shows the cross-correlation function and the resultant displacement vector computed from these images. The resultant displacement vector is  $(5.00, 0.00)$  pixels which is significantly lower than the expected displacement vector of  $(10.0, 0.00)$  pixels. The result shows that the cross-correlation algorithm underestimates the displacement when features are cut as they enter the block.

Figure 62 shows one example of the rectangular feature leaving the interrogation window. Panel a shows the feature is entirely inside the block at  $t_1$ . Panel b shows the image at  $t_2$ . After moving 10 units in the positive x-direction, the feature is cut by the right edge of the block. Panel c shows the cross-correlation function and the resultant displacement vector computed from these blocks. As the result shown in Figure 61, the resultant displacement vector is  $(5.00, 0.00)$  pixels which is significantly lower than the expected displacement vector of  $(10.0, 0.00)$  pixels. The result shows that the cross-correlation algorithm underestimates the

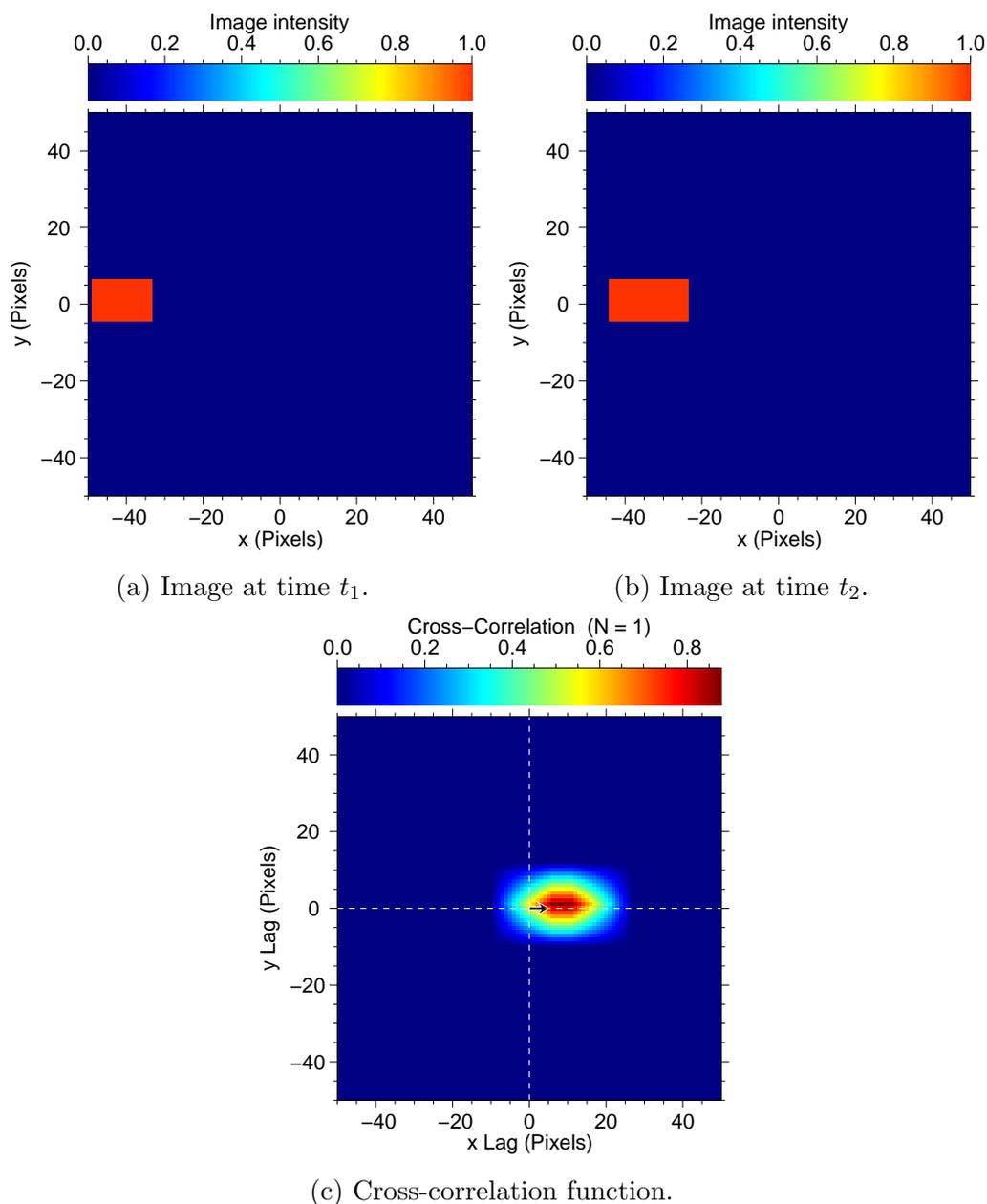


Figure 61. Effects of a feature entering the interrogation window are investigated by using a rectangular pulse of dimensions  $20 \times 10$  pixels moving 10 units in the positive x-direction in time interval  $\Delta t = t_2 - t_1$ . Panel a shows the feature just entering to the interrogation window at  $t_1$ . The feature is cut by an image edge. Panel b shows the image at  $t_2$ . After moving 10 units in the positive x-direction, the feature is in the interrogation window. Panel c shows the the cross-correlation function and the resultant displacement vector applied to these images. The resultant displacement vector is  $(5.00, 0.00)$  pixels which is significantly lower than the expected displacement vector  $(10.0, 0.00)$  pixels.

displacement of features straddling the block edge as they leave the interrogation window.

From these examples, the following observations associated with features straddling and entering or leaving the interrogation window were made:

- (1) the magnitude of the resultant displacement vector is lower than that expected from the velocity field if aerosol features are cut when entering the interrogation window, and
- (2) the magnitude of the resultant displacement vector is lower than that expected from the velocity field if aerosol features are cut when leaving the interrogation window.

These underestimations may not be apparent for point-like particles (as used in PIV) entering or leaving the block edge, since it is less likely they are cut. However, these underestimations must be addressed if the cross-correlation algorithm is applied to lidar backscatter images since aerosol features are larger.

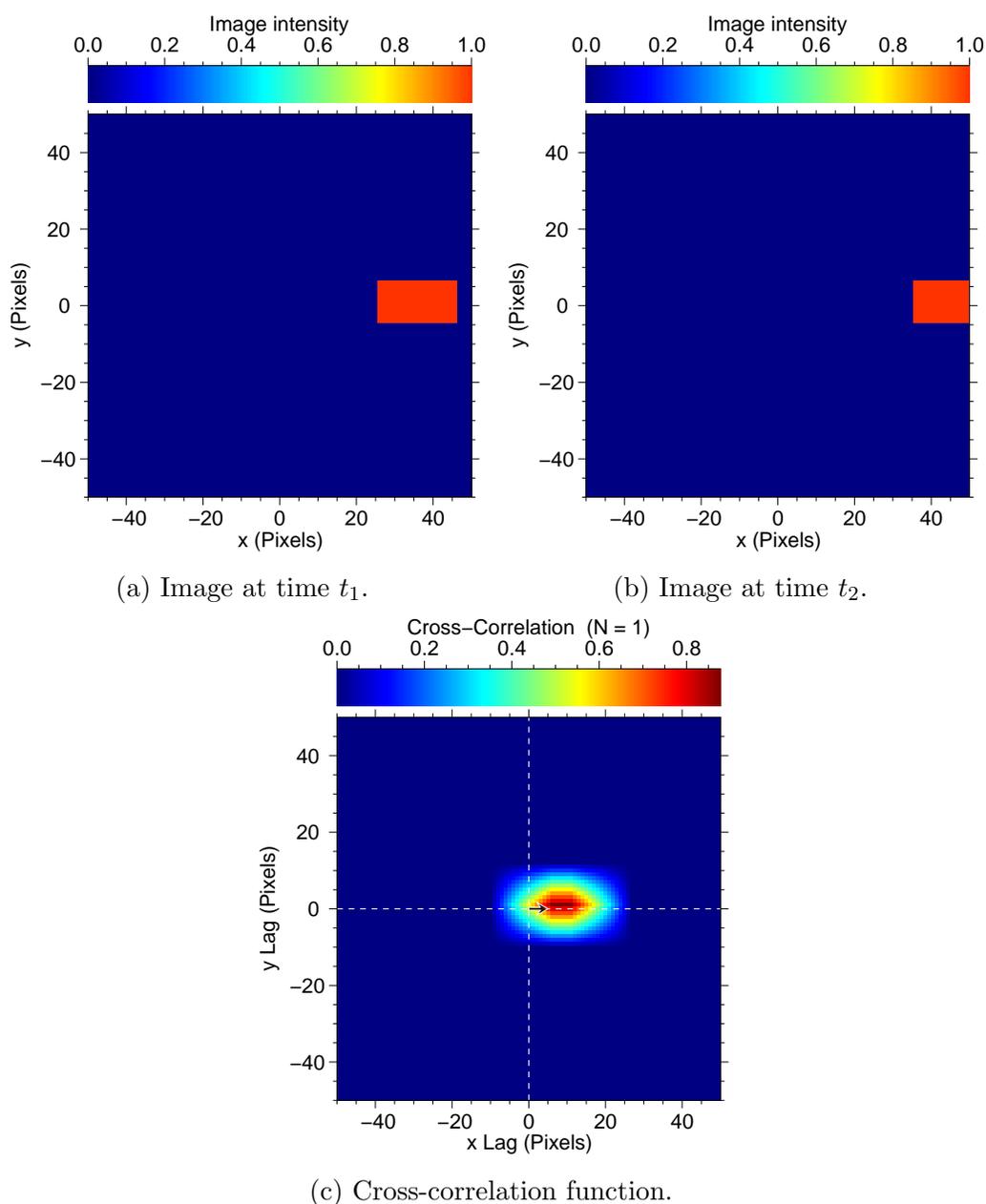


Figure 62. Effects of a feature leaving the interrogation window are investigated by using a rectangular pulse of dimensions  $20 \times 10$  pixels moving 10 units in the positive x-direction in time interval  $\Delta t = t_2 - t_1$ . Panel a shows the feature is in the interrogation window at  $t_1$ . Panel b shows the image at  $t_2$ . After moving 10 units in the positive x-direction, the feature is cut by an image edge. Panel c shows the the cross-correlation function and the resultant displacement vector applied to these images. As the result shown in Figure 61, the resultant displacement vector is  $(5.00, 0.00)$  pixels which is significantly lower than the expected displacement vector  $(10.0, 0.00)$  pixels.

### Application of the Tukey Window

The cross-algorithm is biased toward underestimation of the resultant displacement vector if aerosol features straddle while either entering or leaving the interrogation window. Such underestimation is likely associated with the fact that these features are cut by image edges. To reduce the undesirable effects of these features, one may apply window functions and taper the intensity of features near the block edges. One of many possible window functions that may be applied is the Tukey window. Let  $N$  be the dimension in the x-direction. Then, the 1-D Tukey window  $w(x)$  is defined as

$$w(x) = \begin{cases} \frac{1}{2}(1 + \cos [\pi(\frac{2x}{\alpha(N-1)} - 1)]) & : 0 \leq x \leq \frac{\alpha(N-1)}{2} \\ 1 & : \frac{\alpha(N-1)}{2} \leq x \leq 1(N-1)(1 - \frac{\alpha}{2}) \\ \frac{1}{2}(1 + \cos [\pi(\frac{2x}{\alpha(N-1)} - \frac{2}{\alpha} + 1)]) & : (N-1)(1 - \frac{\alpha}{2}) \leq x \leq (N-1) \end{cases}$$

where  $\alpha$  is a constant (Tukey, 1967). The 1-D Tukey window has a cosine lobe of the width  $\frac{\alpha N}{2}$  that is convolved with a rectangular window of the width  $(1 - \frac{\alpha}{2})N$ . The Tukey window can be extended to two dimensions. In this case, the 2-D Tukey window  $w(x, y)$  is expressed as

$$w(x, y) = \begin{cases} \frac{1}{2}(1 + \cos [\pi(\frac{2x}{\alpha(N-1)} - 1)]) & : 0 \leq x \leq \frac{\alpha(N-1)}{2} \\ 1 & : \frac{\alpha(N-1)}{2} \leq x \leq 1(N-1)(1 - \frac{\alpha}{2}) \\ \frac{1}{2}(1 + \cos [\pi(\frac{2x}{\alpha(N-1)} - \frac{2}{\alpha} + 1)]) & : (N-1)(1 - \frac{\alpha}{2}) \leq x \leq (N-1) \\ \frac{1}{2}(1 + \cos [\pi(\frac{2y}{\alpha(N-1)} - 1)]) & : 0 \leq y \leq \frac{\alpha(N-1)}{2} \\ 1 & : \frac{\alpha(N-1)}{2} \leq y \leq 1(N-1)(1 - \frac{\alpha}{2}) \\ \frac{1}{2}(1 + \cos [\pi(\frac{2y}{\alpha(N-1)} - \frac{2}{\alpha} + 1)]) & : (N-1)(1 - \frac{\alpha}{2}) \leq y \leq (N-1) \end{cases}$$

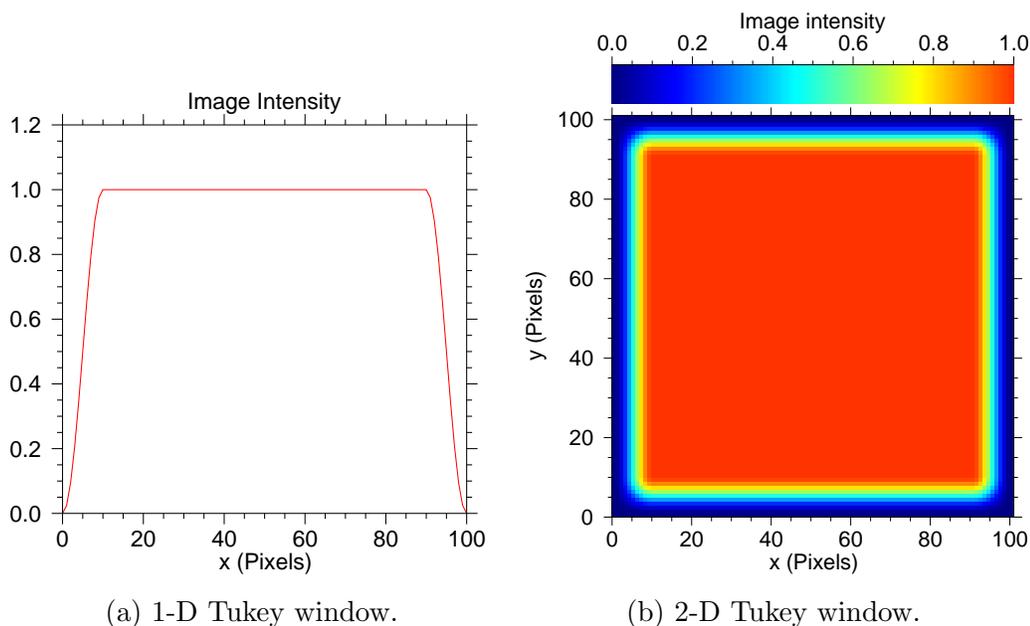


Figure 63. Panels a and b show the 1-D and the 2-D Tukey windows with  $\alpha = 0.20$  and dimensions of 100 pixels.

Figure 63 shows the 1-D and 2-D Tukey windows. The intensity is unity in the middle of the interrogation area and drops to zero at the image edges. The Tukey window preserves the intensity of most aerosol features in the interrogation window, but reduces the intensity of aerosol features near the image edges.

Figures 64 and 65 show the performance of the cross-correlation algorithm for a rectangular pulse straddling and entering or leaving the interrogation window with image edges masked by the Tukey window. These cases are similar to the examples in Figures 61 and 62 respectively. The results show that the resultant displacement vectors for both cases are (9.00, 0.00) pixels. Comparing with corresponding results without masking, (5.00, 0.00) pixels, these displacement vectors are significantly closer to the expected displacement vectors (10.0, 0.00) pixels. Thus, the Tukey window reduces the error of the displacement vector associated with entering or exiting features cut by the image edges.

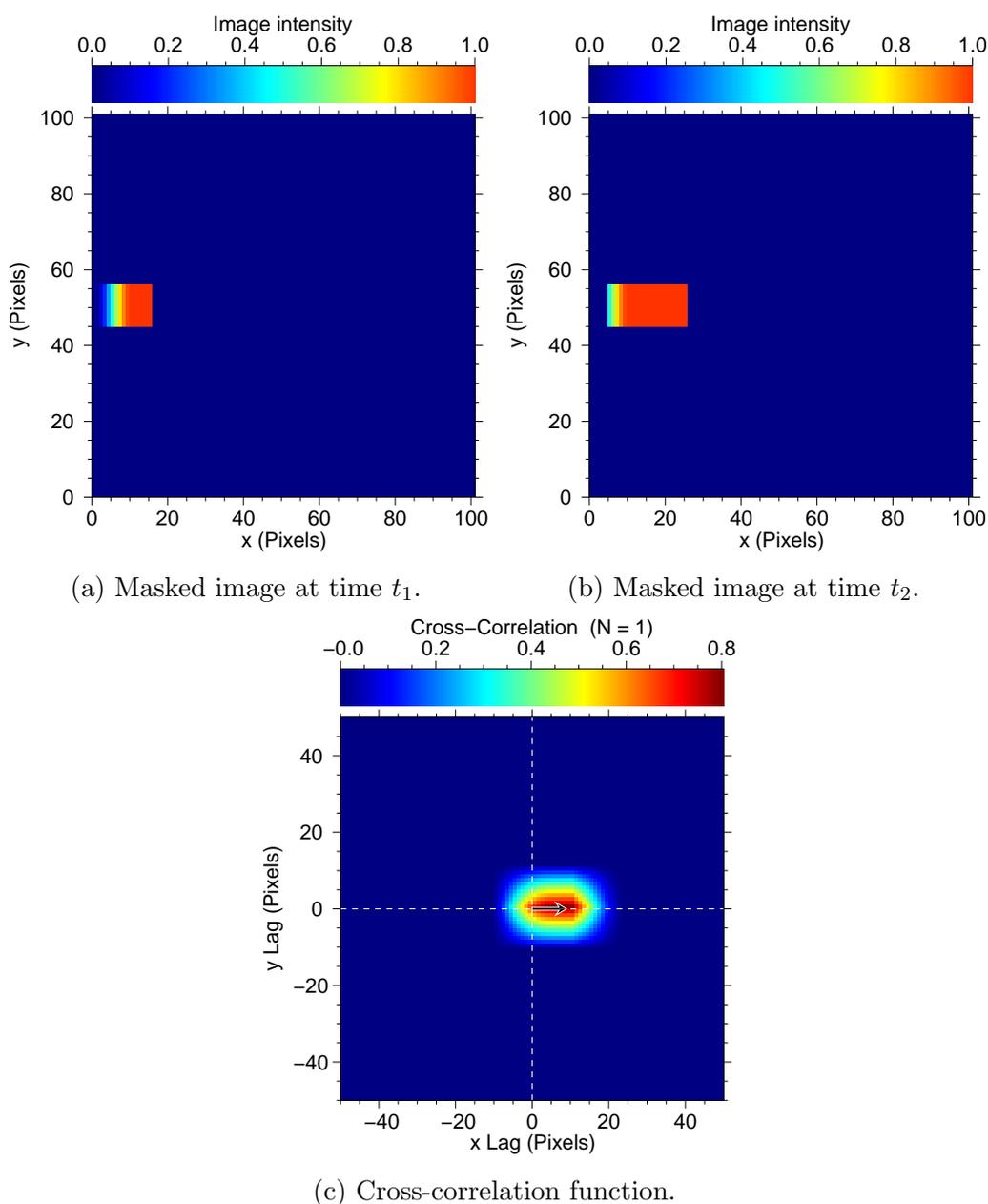
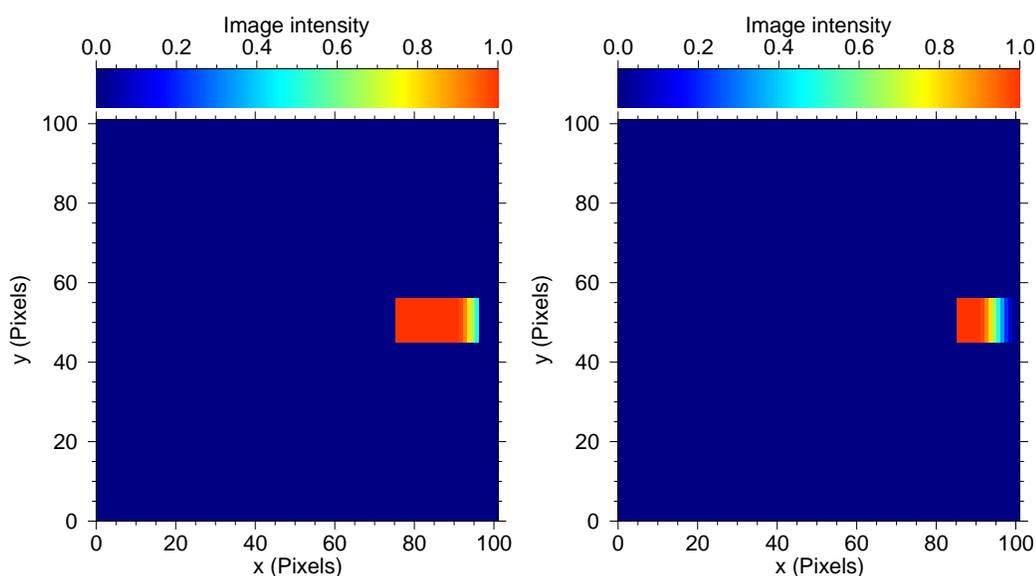
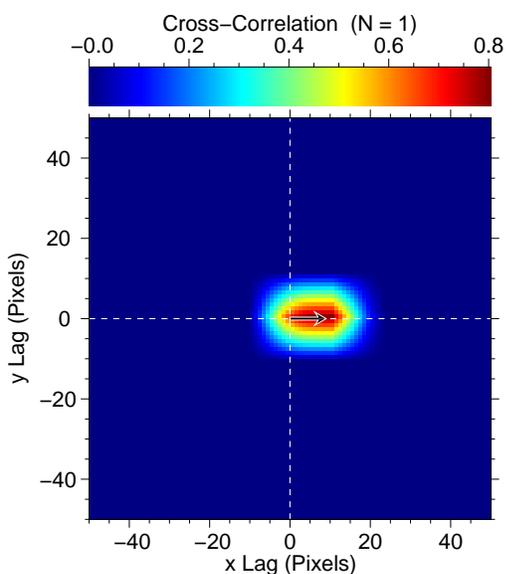


Figure 64. Effects of the Tukey window are investigated by using a rectangular pulse of dimensions  $20 \times 10$  pixels moving 10 units in the positive  $x$ -direction in time interval  $\Delta t = t_2 - t_1$ . Panel a shows the feature entering the interrogation window at  $t_1$ . The feature is cut by an image edge but its intensity near the image edge is reduced by the Turkey window. Panel b shows the image at  $t_2$ . After moving 10 units in the positive  $x$ -direction, the feature is entirely in the interrogation window. Panel c shows the cross-correlation function and the resultant displacement vector applied to these images. The resultant displacement vector is  $(9.00, 0.00)$  pixels which is slightly lower than the expected displacement vector  $(10.0, 0.00)$  pixels, but the performance of the cross-correlation algorithm increases.

(a) Masked image at time  $t_1$ .(b) Masked image at time  $t_2$ .

(c) Cross-correlation function.

Figure 65. Effects of the Tukey window are investigated by using a rectangular pulse of dimensions  $20 \times 10$  pixels moving 10 units in the positive  $x$ -direction in time interval  $\Delta t = t_2 - t_1$ . Panel a shows the feature at  $t_1$ . Panel b shows the image at  $t_2$ . After moving 10 units in the positive  $x$ -direction, the feature is in the interrogation window, the feature is cut by an image edge but its intensity near the image edge is reduced by the Tukey window. Panel c shows the the cross-correlation function and the resultant displacement vector applied to these images. The resultant displacement vector is  $(9.00, 0.00)$  pixels which is slightly lower than the expected displacement vector  $(10.0, 0.00)$  pixels, but much closer than the result without the Tukey window as shown in Figure 61.

### Performance of the Cross-Correlation Algorithm

From the results, one can see that the cross-correlation algorithm gives a perfect displacement vector only if the velocity field is perfectly uniform in the interrogation window and no features straddle the block edges while entering or leaving. However, the performance of the cross-correlation algorithm decreases as the non-uniformity of the velocity field increases because only the global peak of the cross-correlation function is used to give one displacement vector for an entire block. The information contained at non-global peak lags is not used.

For the synthetic data, the shape of the cross-correlation function is not symmetric, and it tends to be weighted more toward lower lags, close to the origin  $(x_{lag}, y_{lag}) = (0, 0)$ . As a result, the magnitudes of the resultant displacement vectors tend to be slightly lower than the ones predicted from the given velocity fields. In addition, the cross-correlation algorithm tends to underestimate the resultant displacement vector if features straddle the block edges while entering or leaving because the shape of the cross-correlation function is distorted by such clipping. Thus, the underestimation of the resultant displacement vectors may be present if one applies the cross-correlation algorithm to estimate the wind velocity from lidar backscatter images. Underestimation of the displacement vector is likely associated with the following observations:

- (1) features with lower velocity tend to appear in both images of two consecutive frames, while those with higher velocity tend to move out from the interrogation window, and
- (2) features straddling and entering or leaving the block edge are sources of undesirable effects of the cross-correlation algorithm.

The observation (1) is likely associated with non uniformity of the velocity field within the interrogation window. The underestimation of the resulting displacement vector due to the observation (1) can be reduced by choosing a smaller interrogation window in which the velocity field may be more uniform. The observation (2) is likely associated with the edge effects. The underestimation of the resulting displacement vector due to the observation (2) can be reduced by using window functions to taper the block edges, and choosing the larger interrogation window where the relative sizes of aerosol features are small and negligible compared with the size of the interrogation window. To maximize the performance of the cross-correlation algorithm, one should choose the size of the interrogation window that compromises the observations (1) and (2).

From the results of tests of the cross-correlation algorithm using synthetic backscatter images, the performance of the cross-correlation algorithm is maximized by:

- (a) using zero-padded images,
- (b) using window functions (i.e., the Tukey window) to eliminate the sharp edges, and
- (c) applying the multiple pass interrogation and curve fitting (i.e., polynomial fitting).

Since the FFT method is computationally more efficient than the covariance method and there is no difference in their results, the FFT method is preferable. The results show that the histogram equalization does not increase the performance of the cross-correlation algorithm, and hence is not recommended. Figure 66 shows the velocity field, created by the sum of the uniform flow and the turbulent perturbation field (the aerodynamic roughness length  $z_0 = 0.5$  m; altitude 30 m), applied to  $N = 100$  pairs of synthetic backscatter images, and the result of the

cross-correlation algorithm with the FFT method, using zero-padded images, tapering by the Tukey window, multiple pass interrogation, and curve fitting. The mean velocity and standard deviation of the given velocity field are (11.08, 0.104) (pixels/frames) and (1.97, 1.23) (pixels/frames), respectively. The mean and standard deviation of the resultant displacement vector, by applying the cross-correlation algorithm described above, are (10.93, 0.0986) (pixels) and (0.816, 0.403) (pixels), respectively. The error of the magnitude of the displacement vector was reduced to 1.35%.

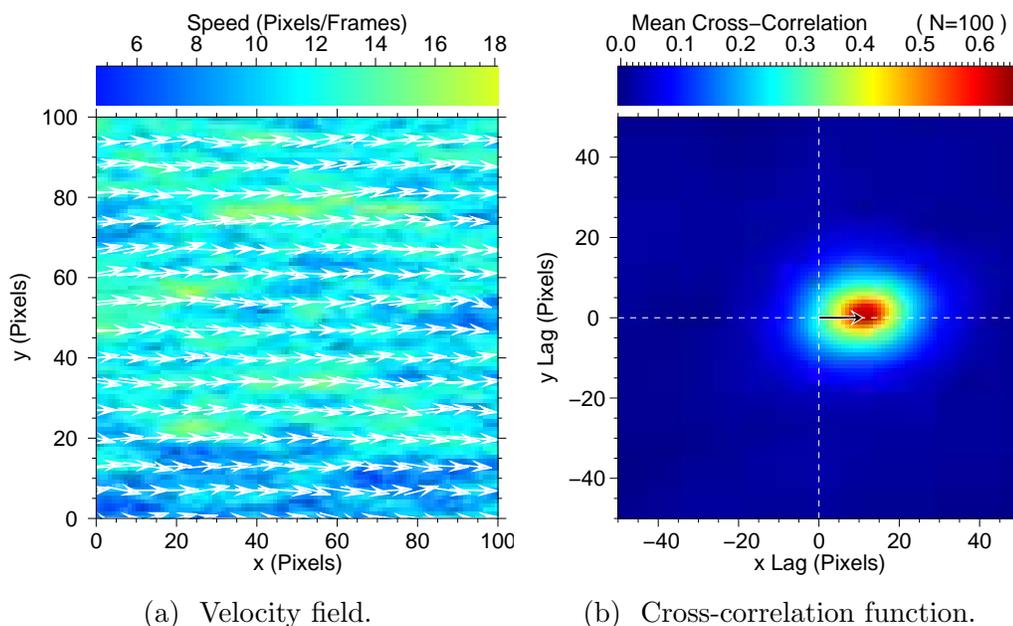


Figure 66. Panel a shows the velocity field, created by the sum of the uniform flow and the turbulent perturbation field, is applied to  $N = 100$  pairs of synthetic backscatter images. The mean velocity and standard deviation are (11.08, 0.104) (pixels/frames) and (1.97, 1.23) (pixels/frames), respectively. Panel b shows the mean cross-correlation function and resultant displacement vector. The mean and standard deviation of the displacement vector are (10.93, 0.0986) (pixels) and (0.816, 0.403) (pixels), respectively.

### Limitation of the Study

This study is based on evaluations of horizontal wind vectors that are 2-D. Real turbulent atmospheric flow is 3-D and a source of false apparent motions which cannot be simulated with 2-D flows. For example, suppose a large spherical aerosol feature moves upward and crosses the horizontal scan plane in the time interval  $\Delta t = t_2 - t_1$ , as shown as Figure 67a. In this case, the 2-dimensional blocks 1 and 2, look like the ones shown in Figure 67b. Here, one sees that the size of the spherical feature appears to increase as the feature passes through the horizontal plane. Then, the application of the cross-correlation algorithm would show divergence. However, the actual size of the feature in 3-D does not change. So, a full error analysis cannot be conducted unless the 3-D space and flow field are taken into account. Since the study does not include the vertical component of the wind vector, no conclusions can be made about the impact of “false apparent motions”.

In addition to the false apparent motions, the real lidar data contains random noise and attenuation. Furthermore, the real lidar data is collected in a polar coordinate system and interpolated to a Cartesian grid before applying motion estimation algorithms. Therefore, the true resolution of the lidar backscatter data decreases with increasing range from the lidar system. The synthetic backscatter fields used in this thesis do not provide the ability to determine the effects of these artifacts.

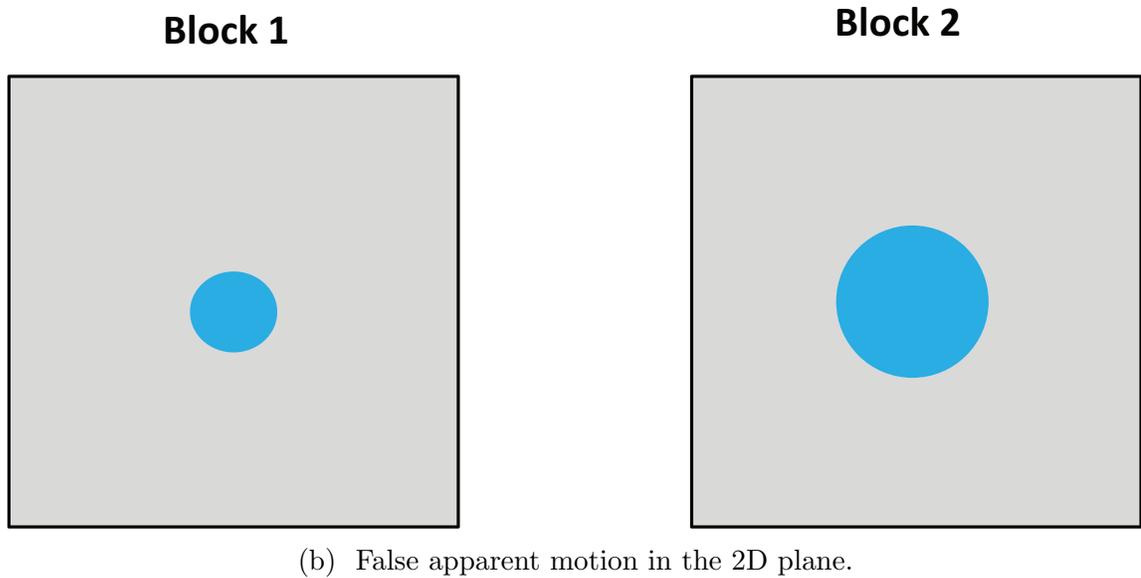
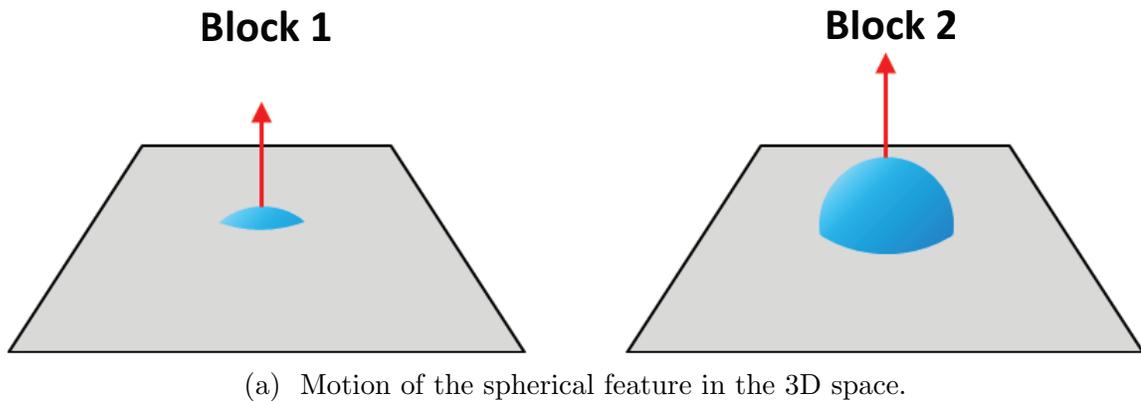


Figure 67. False apparent motion of the spherical feature. Panel a represents the upward motion of the spherical feature. Panel b represents the motion of the spherical feature in the 2-dimensional plane. The size of the feature appears to be increased.

## Conclusions

The cross-correlation approach to deriving motion vectors from image sequences is widely used in science and engineering. The most widely used form of cross-correlation is PIV in which the flow is seeded with tiny particles. These tracers are distributed approximately uniformly throughout the image and their discrete nature results in cross-correlation functions with relatively sharp peaks and low ambiguity. Cross-correlation applied to atmospheric aerosol lidar images results in broad peaks and increased ambiguity because the tracer of the flow is a continuous field of aerosol backscatter intensity that changes shape and brightness in response to the flow. The cross-correlation algorithm was applied several times previously to real lidar data and several attempts were made to compare the results with other forms of wind measurements. However, an evaluation of the cross-correlation algorithm using synthetic aerosol backscatter images and synthetic velocity fields had not been performed.

The results show that the cross-correlation algorithm gives perfect displacement vectors if the flow is uniform in the interrogation window, and no aerosol features straddle the edge of the blocks while entering or leaving the interrogation windows. However, the performance of the cross-correlation algorithm decreases as the flow fields become non-uniform, and aerosol features straddle the block edges. It was also found that the magnitudes of the resultant displacement vectors tend to be slightly lower than the means computed from the corresponding velocity fields. Such underestimation may be attributed to the fact that features with lower velocity tend to appear in both images of two consecutive frames, while those that have a higher velocity tend to move out of the interrogation window. In addition, edge straddling features are also sources of velocity underestimation.

Finally, it was also found that both the area and the intensity of features do not influence the performance of the cross-correlation algorithm equally. It was found that the intensity of the features contributes more than a factor of four compared to the equivalent area.

The results show that using zero-padded input arrays, tapering the block edges with the Tukey window, and applying the multiple pass interrogation and curve fitting (i.e., polynomial fitting) can be used to minimize the underestimation of the resultant vector. By maximizing the performance of the cross-correlation algorithm, the mean resultant displacement vector for moderate turbulent flows without large coherent flow structures was found to be within a few percent of the one expected from the corresponding velocity fields. A study involving 3-D velocity fields and aerosol structures, and that mimics some of the artifacts of real lidar data, is suggested for a more comprehensive understanding.

## REFERENCES

## REFERENCES

- Adrian, R. J. and Westerweel, J. (2011). *Particle Image Velocimetry*. Cambridge University Press.
- Antoine, E., Buchanan, C., Fezzaa, K., Lee, W.-K., Rylander, M. N., and Vlachos, P. (2013). Flow measurements in a blood-perfused collagen vessel using x-ray micro-particle image velocimetry. *PLOS ONE*, 8(11):e81198.
- Arfken, G. B. and Weber, H.-J. (2001). *Mathematical Methods for Physicists*. Harcourt/Academic Press.
- Arya, S. P. (1999). *Air Pollution Meteorology and Dispersion*. Oxford University Press.
- Bailey, B. H., Beaucage, P., Bernadett, D. W., and Brower, M. (2012). *Wind Resource Assessment: A Practical Guide to Developing a Wind Project*. John Wiley & Sons.
- Bailey, B. H., McDonald, S. L., Bernadett, D. W., Markus, M. J., and Elsholz, K. V. (1997). Wind Resource Assessment Handbook: Fundamentals for Conducting a Successful Monitoring Program. Technical report, National Renewable Energy Lab., Golden, CO (US); AWS Scientific, Inc., Albany, NY (US).
- Baker, W. E. and Coauthors (2013). Lidar-measured wind profiles: The missing link in the global observing system. *Bull. Amer. Meteor. Soc.*, 95:543–564.

- Bastiaans, R. J. (2000). *Cross-correlation PIV: Theory, Implementation and Accuracy*. Eindhoven University of Technology, Faculty of Mechanical Engineering.
- Bendat, J. S. and Piersol, A. G. (2011). *Random Data: Analysis and Measurement Procedures*. John Wiley & Sons.
- Burton, T., Jenkins, N., Sharpe, D., and Bossanyi, E. (2011). *Wind Energy Handbook*. John Wiley & Sons.
- Cheng, W., Murai, Y., Sasaki, T., and Yamamoto, F. (2005). Bubble velocity measurement with a recursive cross correlation PIV technique. *Flow Meas. Instrum.*, 16(1):35–46.
- Davenport, A. G., Grimmond, C. S. B., Oke, T. R., and Wieringa, J. (2000). Estimating the roughness of cities and sheltered country. In *15th Conference on Probability and Statistics in the Atmospheric Sciences/12th Conference on Applied Climatology, Ashville, NC, Amer. Meteor. Soc.*, pages 96–99.
- Davis, J. C. and Sampson, R. J. (2002). *Statistics and Data Analysis in Geology*. Wiley.
- Debella-Gilo, M. and Kääh, A. (2011). Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation. *Remote Sens. Environ.*, 115(1):130–142.
- Debella-Gilo, M. and Kääh, A. (2012). Measurement of surface displacement and deformation of mass movements using least squares matching of repeat high resolution satellite and aerial images. *Remote Sens.*, 4(1):43–67.

- Eloranta, E. W., King, J. M., and Weinman, J. A. (1975). The determination of wind speeds in the boundary layer by monostatic lidar. *J. Appl. Meteor.*, 14:1485–1489.
- Gentry, B. M., Chen, H., and Li, S. X. (2000). Wind measurements with 355-nm molecular Doppler lidar. *Opt. Lett.*, 25(17):1231–1233.
- Gentry, B. M. and Korb, C. L. (1994). Edge technique for high-accuracy Doppler velocimetry. *Appl. Optics*, 33(24):5770–5777.
- Grund, C. and Tucker, S. C. (2011). Optical autocovariance wind lidar (OAWL): A new approach to direct-detection Doppler wind profiling. In *5th Symp. on Lidar Atmospheric Applications*, Seattle, WA. Amer. Meteor. Soc.
- Grund, C. J., Banta, R. M., George, J. L., Howell, J. N., Post, M. J., Richter, R. A., and Weickman, A. M. (2001). High-resolution Doppler lidar for boundary layer and cloud research. *J. Atmos. Ocean. Technol.*, 18:376–393.
- Henderson, S. W., Hale, C. P., Magee, J. R., Kavaya, M. J., and Huffaker, A. V. (1991). Eye-safe coherent laser radar system at 2.1 micron using Tm,Ho:YAG lasers. *Opt. Lett.*, 16:773–775.
- Hooper, W. P. and Eloranta, E. W. (1986). Lidar measurements of wind in the planetary boundary layer: the method, accuracy and results from joint measurements with radiosonde and kytoon. *J. Clim. Appl. Meteor.*, 25:990–1001.
- Kanak, K. M. (2005). Numerical simulation of dust devil-scale vortices. *Quart. J. R. Met. Soc.*, 131(607):1271–1292.
- Keane, R. D. and Adrian, R. J. (1992). Theory of cross-correlation analysis of PIV images. *Appl. Sci. Res.*, 49(3):191–215.

- Kolev, I., Parvanov, O., and Kaprielov, B. (1988). Lidar determination of winds by aerosol inhomogeneities: motion velocity in the planetary boundary layer. *Appl. Optics*, 27:2524–2531.
- Köpp, F., Schwiesow, R., and Werner, C. (1984). Remote measurements of boundary-layer wind profiles using a cw Doppler lidar. *J. Clim. Appl. Meteor.*, 23(1):148–154.
- Kunkel, K. E., Eloranta, E. W., and Weinman, J. (1980). Remote determination of winds, turbulence spectra and energy dissipation rates in the boundary layer from lidar measurements. *J. Atmos. Sci.*, 37:978–985.
- Leese, J. A., Novak, C. S., and Clark, B. B. (1971). An automated technique for obtaining cloud motion from geosynchronous satellite data using cross correlation. *J. Appl. Meteor.*, 10:118–132.
- Mann, J. (1994). The spatial structure of neutral atmospheric surface-layer turbulence. *J. Fluid Mech*, 273:141–168.
- Mann, J. (1998). Wind field simulation. *Prob. Engng. Mech*, 13:269–282.
- Mann, J., Peña, A., Bingol, F., Wagner, R., and Courtney, M. S. (2010). Lidar scanning of momentum flux in and above the atmospheric surface layer. *J. Atmos. Ocean. Technol.*, 27:959–976.
- Mathews, J. and Walker, R. L. (1970). *Mathematical Methods of Physics*. W. A. Benjamin.
- Mauzey, C. F., Lowe, J. P., and Mayor, S. D. (2012). Real-time wind velocity estimation from aerosol lidar data using graphics hardware. GPU Technology Conference, San Jose, CA. Poster presentation AV10.

- Mayor, S. D. (2011). Observations of seven density current fronts in Dixon, California. *Mon. Wea. Rev.*, 139:1338–1351.
- Mayor, S. D., Dérian, P., Mauzey, C. F., and Hamada, M. (2013). Two-component wind fields from scanning aerosol lidar and motion estimation algorithms. SPIE. Paper 9972-7.
- Mayor, S. D. and Eloranta, E. W. (2001). Two-dimensional vector wind fields from volume imaging lidar data. *J. Appl. Meteor.*, 40:1331–1346.
- Mayor, S. D., Lowe, J. P., and Mauzey, C. F. (2012). Two-component horizontal aerosol motion vectors in the atmospheric surface layer from a cross-correlation algorithm applied to scanning elastic backscatter lidar data. *J. Atmos. Ocean. Technol.*, 29:1585–1602.
- Mayor, S. D. and Spuler, S. M. (2004). Raman-shifted Eye-safe Aerosol Lidar. *Appl. Optics*, 43:3915–3924.
- Mayor, S. D., Spuler, S. M., Morley, B. M., and Loew, E. (2007). Polarization lidar at 1.54-microns and observations of plumes from aerosol generators. *Opt. Eng.*, 46:DOI: 10.1117/12.781902.
- McGill, M. J., Skinner, W. R., and Irgang, T. D. (1997). Analysis techniques for the recovery of winds and backscatter coefficients from a multiple-channel incoherent Doppler lidar. *Appl. Opt.*, 36(6):1253–1268.
- Middleton, W. E. K. (1969). *Invention of the Meteorological Instruments*. The Johns Hopkins Press.

- Newsom, R. K., Ligon, D., Calhoun, R., Heap, R., Cregan, E., and Princevac, M. (2005). Retrieval of microscale wind and temperature fields from single- and dual-Doppler lidar data. *J. Appl. Meteor.*, 44:1324–1345.
- Pearson, G., Davies, F., and Collier, C. (2009). An analysis of the performance of the UFAM pulsed Doppler lidar for observing the boundary layer. *J. Atmos. Ocean. Technol.*, 26:240–250.
- Piironen, A. K. and Eloranta, E. W. (1995). Accuracy analysis of wind profiles calculated from volume imaging lidar data. *J. Geophys. Res.*, 100:25559–25567.
- Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press.
- Post, M. J. and Cupp, R. E. (1990). Optimizing a pulsed Doppler lidar. *Appl. Optics*, 29:4115–4158.
- Raffel, M. (2007). *Particle Image Velocimetry: A Practical Guide*. Springer.
- Reitebuch, O., Werner, C., Leike, I., Delville, P., Flamant, P. H., Cress, A., and Engelbart, D. (2001). Experimental validation of wind profiling performed by the airborne 10- $\mu\text{m}$  heterodyne Doppler lidar WIND. *J. Atmos. Ocean. Technol.*, 18(8):1331–1344.
- Sasano, Y., Hirohara, H., Yamasaki, T., Shimizu, H., Takeuchi, N., and Kawamura, T. (1982). Horizontal wind vector determination from the displacement of aerosol distribution patterns observed by a scanning lidar. *J. Appl. Meteor.*, 21(10):1516–1523.
- Sathe, A. and Mann, J. (2012). Measurement of turbulence spectra using scanning pulsed wind lidars. *J. Geophys. Res.*, 117:D01201, DOI:10.1029/2011JD016786.

- Sathe, A., Mann, J., Gottschall, J., and Courtney, M. S. (2011). Can wind lidars measure turbulence? *J. Atmos. Ocean. Technol.*, 28:853–868.
- Schmidt, H. and Schumann, U. (1989). Coherent structures of the convective boundary layer derived from large-eddy simulations. *J. Fluid Mech.*, 200:511–562.
- Schols, J. L. and Eloranta, E. W. (1992). The calculation of area-averaged vertical profiles of the horizontal wind velocity from volume imaging lidar data. *J. Geophys. Res.*, 97:18395–18407.
- Schubert, A., Faes, A., Kääh, A., and Meier, E. (2013). Glacier surface velocity estimation using repeat TerraSAR-X images: Wavelet-vs. correlation-based image matching. *Int. Soc. Photogramme.*, 82:49–62.
- Schubert, A., Jehle, M., Small, D., and Meier, E. (2010). Influence of atmospheric path delay on the absolute geolocation accuracy of TerraSAR-X high-resolution products. *IEEE Trans. Geosci. Remote Sensing*, 48(2):751–758.
- Shimizu, H., Sasano, Y., Yasuoka, Y., Ueda, H., Takeuchi, N., and Okuda, M. (1981). The remote measurement of wind direction and velocity by a laser radar using the spatial correlation technique. *Oyobutsuri*, 50:616–620.
- Spuler, S. M. and Mayor, S. D. (2005). Scanning eye-safe elastic backscatter lidar at 1.54 microns. *J. Atmos. Ocean. Technol.*, 22:696–703.
- Spuler, S. M. and Mayor, S. D. (2007). Raman shifter optimized for lidar at 1.5-micron wavelength. *Appl. Optics*, 46:2990–2995.
- Sroga, J. T., Eloranta, E. W., and Barber, T. (1980). Lidar measurements of wind velocity profiles in the boundary layer. *J. Appl. Meteor.*, 19:598–605.

- Stull, R. B. (1988). *An Introduction to Boundary Layer Meteorology*. Kluwer, Boston.
- Telford, W. M., Geldart, L. P., and Sheriff, R. E. (1990). *Applied Geophysics*. Cambridge University Press.
- Tukey, J. W. (1967). *An introduction to the calculations of numerical spectrum analysis*, pages 25–46. John Wiley and Sons.
- Venditti, J. G., Best, J. L., Church, M., and Hardy, R. J. (2013). *Coherent Flow Structures at Earth's Surface*. Wiley, UK, 1st edition.
- Yun, S., Ni, S., Park, M., and Lee, W. S. (2009). Southeast Indian ocean-ridge earthquake sequences from cross-correlation analysis of hydroacoustic data. *Geophys. J. Int.*, 179(1):401–407.

## APPENDIX A

## APPENDIX A

### THE CROSS-CORRELATION FUNCTION FOR A GAUSSIAN FEATURE

#### Analytical Approaches Using a Gaussian Feature

##### Gaussian Integrals

The cross-correlation functions for lidar backscatter images are usually calculated numerically by the fast Fourier transform (FFT). However, for some simple cases, one can find the cross-correlation function analytically, and find exact solution of the resultant displacement vector. Here, analytical methods using Gaussian features are introduced. Backscatter intensities of such features can be expressed by a Gaussian function,  $e^{-\lambda x^2}$ , where  $\lambda$  is a constant. To calculate the cross-correlation functions of such backscatter images, one needs to evaluate the following integral.

$$I = \int_{-\infty}^{\infty} e^{-\lambda x^2} dx. \quad (1)$$

This type of integrals can be evaluated as follows. First, squaring both sides of this equation gives

$$I^2 = \left[ \int_{-\infty}^{\infty} e^{-\lambda x^2} dx \right] \left[ \int_{-\infty}^{\infty} e^{-\lambda x^2} dx \right]. \quad (2)$$

Next, switch  $x$  and  $y$  for the second integral.

$$I^2 = \left[ \int_{-\infty}^{\infty} e^{-\lambda x^2} dx \right] \left[ \int_{-\infty}^{\infty} e^{-\lambda y^2} dy \right]. \quad (3)$$

This integral can be expressed as the double integral in the  $xy$ -plane.

$$I^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\lambda(x^2+y^2)} dx dy. \quad (4)$$

The limits of integration and the area element in the Cartesian coordinates are shown in Figure 68a. Now, one can rewrite this integral in polar coordinates,  $\rho = \sqrt{x^2 + y^2}$ , and  $\phi = \tan^{-1}(\frac{y}{x})$ . In this case, the area element becomes  $dx dy = \rho d\rho d\phi$ , and the limits become  $0 < \rho < \infty$ , and  $0 < \phi < 2\pi$ . The limits of integration and the area element in the polar coordinates are shown in Figure 68b. Using the polar coordinates,  $(\rho, \phi)$ , the integral becomes

$$I^2 = \int_0^{2\pi} \int_0^{\infty} e^{-\lambda \rho^2} \rho d\rho d\phi. \quad (5)$$

The  $\rho$  integral can be evaluated by letting  $u = -\lambda \rho^2$  and  $du = -2\lambda \rho d\rho$ , ( $-\frac{1}{2\lambda} du = \rho d\rho$ ). Then, one has

$$I^2 = -\frac{1}{2\lambda} \int_0^{2\pi} \int_{\rho=0}^{\rho \rightarrow \infty} e^u du d\phi \quad (6)$$

and evaluating  $u$  integral gives

$$I^2 = -\frac{1}{2\lambda} \int_0^{2\pi} [e^u]_{\rho=0}^{\rho \rightarrow \infty} d\phi \quad (7)$$

that is,

$$I^2 = -\frac{1}{2\lambda} \int_0^{2\pi} [e^{-\lambda\rho^2}] \Big|_{\rho=0}^{\rho \rightarrow \infty} d\phi \quad (8)$$

$$I^2 = -\frac{1}{2\lambda} \int_0^{2\pi} [e^{-\infty} - e^0] d\phi \quad (9)$$

$$I^2 = -\frac{1}{2\lambda} \int_0^{2\pi} [0 - 1] d\phi \quad (10)$$

$$I^2 = \frac{1}{2\lambda} \int_0^{2\pi} d\phi. \quad (11)$$

The  $\phi$  integral is  $\int_0^{2\pi} d\phi = 2\pi$ . Thus one has

$$I^2 = \frac{1}{2\lambda} 2\pi \quad (12)$$

or

$$I^2 = \frac{\pi}{\lambda}. \quad (13)$$

Finally, taking square roots both sides of this equation, one can get

$$I = \sqrt{\frac{\pi}{\lambda}} \quad (14)$$

that is,

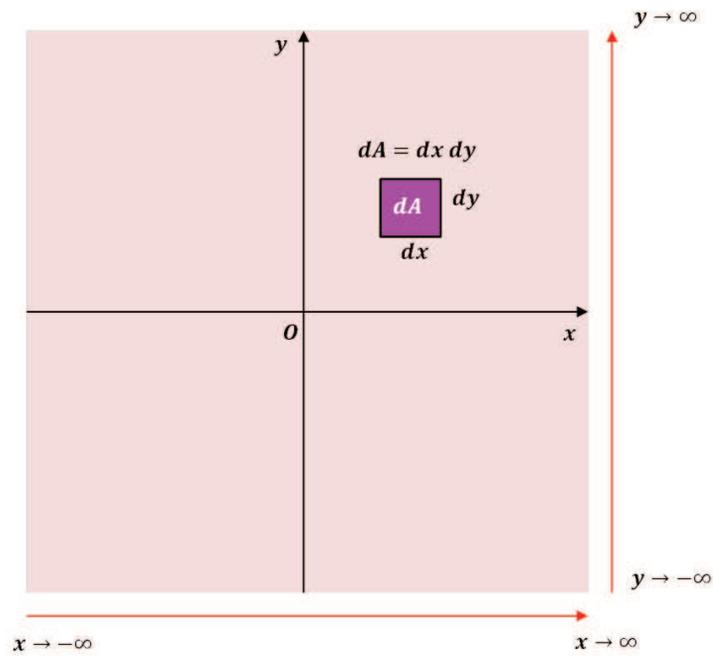
$$I = \int_{-\infty}^{\infty} e^{-\lambda x^2} dx = \sqrt{\frac{\pi}{\lambda}}. \quad (15)$$

Suppose the Gaussian function is instead,  $e^{-\lambda(x-c)^2}$ , where  $c$  is a constant. Then, the position of the peak is shifted  $c$  units from the origin. If  $c$  is positive, the peak is shifted to the positive  $x$  direction. On the other hand, if  $c$  is negative, the peak is

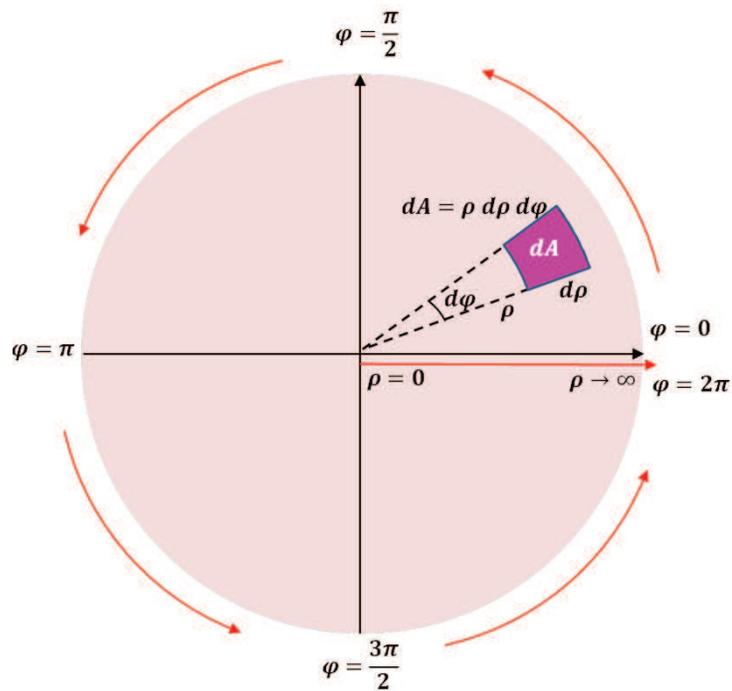
shifted to the negative  $x$  direction. However, the size and the shape of the Gaussian function is unaltered, so the result of the integral must be the same. That is,

$$\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}. \quad (16)$$

This equation is also helpful to find the cross-correlation function analytically.



(a) The limits of integration and the area element in the 2-D plane (Cartesian coordinates).



(b) The limits of integration and the area element in the 2-D plane (polar coordinates).

Figure 68. Limits of integration in the 2-D Cartesian and polar coordinates.

One-Dimensional Case in  
an Uniform Flow

Suppose that there is only one Gaussian feature in a 1-D space, and the Gaussian feature is at the origin,  $x = 0$ , at time  $t_1$ . Then, the backscatter image at time  $t_1$ ,  $f_1(x)$ , can be expressed as

$$f_1(x) = I_0 e^{-x^2}, \quad (17)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Suppose further that the Gaussian feature moves 10 units to the right in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image at time  $t_2$ ,  $f_2(x)$ , can be expressed as

$$f_2(x) = I_0 e^{-(x-10)^2}, \quad (18)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Then, the 1-D cross-correlation function,  $r_x$ , applied to these images can be calculated as follows. The Fourier transform of  $f_1 = I_0 e^{-x^2}$ ,  $F_1(k)$  is

$$F_1(k) = \int_{-\infty}^{\infty} I_0 e^{-x^2} e^{-i2\pi kx} dx \quad (19)$$

or

$$F_1(k) = \int_{-\infty}^{\infty} I_0 e^{-(x^2 + i2\pi kx)} dx. \quad (20)$$

To evaluate this integral, we add and subtract  $(i\pi k)^2$  to the power of e and complete the square. Then, one has

$$F_1(k) = \int_{-\infty}^{\infty} I_0 e^{-[x^2 + i2\pi kx + (i\pi k)^2 - (i\pi k)^2]} dx. \quad (21)$$

After simplifying it gives

$$F_1(k) = I_0 e^{-\pi^2 k^2} \int_{-\infty}^{\infty} e^{-(x+i\pi k)^2} dx. \quad (22)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}$  for any constant  $c$ , the integral  $\int_{-\infty}^{\infty} e^{-(x+i\pi k)^2} dx = \sqrt{\frac{\pi}{\lambda}}$ , where  $\lambda = 1$ . Thus,  $F_1(k)$  can be expressed as

$$F_1(k) = I_0 e^{-\pi^2 k^2} \sqrt{\frac{\pi}{1}} \quad (23)$$

or

$$F_1(k) = \sqrt{\pi} I_0 e^{-\pi^2 k^2}. \quad (24)$$

Next, the Fourier transform of  $f_2 = I_0 e^{-(x-10)^2}$ ,  $F_2(k)$  is

$$F_2(k) = \int_{-\infty}^{\infty} I_0 e^{-(x-10)^2} e^{-i2\pi kx} dx. \quad (25)$$

or, by expanding the power of  $e$  gives

$$F_2(k) = \int_{-\infty}^{\infty} I_0 e^{-(x^2-20x+100+i2\pi kx)} dx \quad (26)$$

$$F_2(k) = I_0 \int_{-\infty}^{\infty} e^{-[x^2+(i2\pi k-20)x+100]} dx. \quad (27)$$

As  $F_1(k)$ ,  $F_2(k)$  can be calculated by adding and subtracting  $(i\pi k - 10)^2$  to the power of e.

$$F_2(k) = I_0 \int_{-\infty}^{\infty} e^{-[x^2 + (i2\pi k - 20)x + 100 + (i\pi k - 10)^2 - (i\pi k - 10)^2]} dx. \quad (28)$$

After simplifying it gives

$$F_2(k) = I_0 \int_{-\infty}^{\infty} e^{-(x+i\pi k-10)^2} e^{-100} e^{(i\pi k-10)^2} dx \quad (29)$$

$$F_2(k) = I_0 e^{-100} e^{(i\pi k-10)^2} \int_{-\infty}^{\infty} e^{-(x+i\pi k-10)^2} dx. \quad (30)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}$  for any constant c, the integral  $\int_{-\infty}^{\infty} e^{-(x+i\pi k-10)^2} dx = \sqrt{\frac{\pi}{\lambda}}$ , where  $\lambda = 1$ . Thus,  $F_2(k)$  becomes

$$F_2(k) = I_0 e^{-100} e^{(i\pi k-10)^2} \sqrt{\frac{\pi}{1}} \quad (31)$$

or

$$F_2(k) = \sqrt{\pi} I_0 e^{-100} e^{(i\pi k-10)^2}. \quad (32)$$

Now taking complex conjugate of  $F_2(k)$  gives

$$F_2^*(k) = \sqrt{\pi} I_0 e^{-100} e^{(-i\pi k-10)^2} \quad (33)$$

or

$$F_2^*(k) = \sqrt{\pi} I_0 e^{-100} e^{(i\pi k+10)^2}. \quad (34)$$

The cross-correlation function  $r_x$  is defined as

$$r_x = F^{-1} \left[ \frac{F_1(k)F_2^*(k)}{S_1S_2} \right] \quad (35)$$

or

$$r_x = \int_{-\infty}^{\infty} \frac{F_1(k)F_2^*(k)e^{i2\pi kx}}{S_1S_2} dk, \quad (36)$$

where  $F_1(k)$  is the Fourier transform of  $f_1(x)$ ,  $F_2^*(k)$  the complex conjugate of the Fourier transform of  $f_2(x)$ ,  $S_1$  the standard deviation of  $f_1(x)$ ,  $S_2$  the standard deviation of  $f_2(x)$ , and  $F^{-1}$  represents the inverse Fourier transform. Substituting  $F_1(k) = \sqrt{\pi}I_0e^{-\pi^2k^2}$  and  $F_2^*(k) = \sqrt{\pi}I_0e^{-100}e^{(i\pi k+10)^2}$  into this equation, one has

$$r_x = \frac{1}{S_1S_2} \int_{-\infty}^{\infty} \sqrt{\pi}I_0e^{-\pi^2k^2} \sqrt{\pi}I_0e^{-100}e^{(i\pi k+10)^2} e^{i2\pi kx} dk \quad (37)$$

$$r_x = \frac{\pi I_0^2}{S_1S_2} e^{-100} \int_{-\infty}^{\infty} e^{-\pi^2k^2} e^{(i\pi k+10)^2} e^{i2\pi kx} dk \quad (38)$$

or, by expanding the power of e and rearranging gives

$$r_x = \frac{\pi I_0^2}{S_1S_2} e^{-100} e^{100} \int_{-\infty}^{\infty} e^{-2\pi^2[k^2 - \frac{i}{\pi}(10+x)k]} dk. \quad (39)$$

The term  $e^{-100}e^{100} = e^0 = 1$ . In addition, adding and subtracting  $[\frac{i}{2\pi}(10+x)k]^2$  to the power of e gives

$$r_x = \frac{\pi I_0^2}{S_1S_2} \int_{-\infty}^{\infty} e^{-2\pi^2(k^2 - \frac{i}{\pi}(10+x)k + [\frac{i}{2\pi}(10+x)k]^2 - [\frac{i}{2\pi}(10+x)k]^2)} dk. \quad (40)$$

After simplifying it gives

$$r_x = \frac{\pi I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{2}} \int_{-\infty}^{\infty} e^{-2\pi^2 [k - \frac{i}{2\pi}(x+10)]^2} dk. \quad (41)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(k-c)^2} dk = \sqrt{\frac{\pi}{\lambda}}$  for any constant  $c$ , the integral  $\int_{-\infty}^{\infty} e^{-2\pi^2 [k - \frac{i}{2\pi}(x+10)]^2} dk = \sqrt{\frac{\pi}{\lambda}}$ , where  $\lambda = 2\pi^2$ . Thus,

$$r_x = \frac{\pi I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{2}} \sqrt{\frac{\pi}{2\pi^2}}. \quad (42)$$

After simplifying it gives

$$r_x = \sqrt{\frac{\pi}{2}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{2}}. \quad (43)$$

Finally, defining  $x_{lag} = -x$  for the lag, the cross-correlation function  $r_{x_{lag}}$  can be expressed as

$$r_{x_{lag}} = \sqrt{\frac{\pi}{2}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(-x_{lag}+10)^2}{2}} \quad (44)$$

or

$$r_{x_{lag}} = \sqrt{\frac{\pi}{2}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x_{lag}-10)^2}{2}}, \quad (45)$$

where  $I_0$  is the peak backscatter intensity of the given Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x)$ , and  $S_2$  is the standard deviation of  $f_2(x)$ .

### Two-Dimensional Case in an Uniform Flow

One can extend the concepts of the 1-D case, described above, to the 2-D case.

Suppose that there is only 1 Gaussian feature in a 2-D space, and the Gaussian feature is at the origin,  $(x, y) = (0, 0)$ , at time  $t_1$ . Then, the backscatter image at

time  $t_1$ ,  $f_1(x, y)$ , can be expressed as

$$f_1(x, y) = I_0 e^{-(x^2+y^2)}, \quad (46)$$

where  $I_0$  is the peak backscatter image of the Gaussian feature. Suppose further that the Gaussian feature moves 10 units to the right (positive x direction) and 10 units upward (positive y direction) in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image at time  $t_2$ ,  $f_2(x, y)$ , can be expressed as

$$f_2(x, y) = I_0 e^{-[(x-10)^2+(y-10)^2]}, \quad (47)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Now, the 2-D cross-correlation function,  $r_{x,y}$ , applied to these images can be calculated as follows. The Fourier transform of  $f_1(x, y) = I_0 e^{-(x^2+y^2)}$ ,  $F_1(k_x, k_y)$  is

$$F_1(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (48)$$

$$F_1(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_0 e^{-(x^2+y^2)} e^{-i2\pi(k_x x + k_y y)} dx dy. \quad (49)$$

To evaluate this double integral analytically, one can make complete squares in the power of e, as we did for the 1-D case. Then, the equation above can be simplified as

$$F_1(k_x, k_y) = I_0 e^{(i\pi k_x)^2} e^{(i\pi k_y)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(y+i\pi k_y)^2} e^{-(x+i\pi k_x)^2} dx dy. \quad (50)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}$  and  $\int_{-\infty}^{\infty} e^{-\lambda(y-c)^2} dy = \sqrt{\frac{\pi}{\lambda}}$  for any constant  $c$ , the integral

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(y+i\pi k_y)^2} e^{-(x+i\pi k_x)^2} dx dy = \left(\sqrt{\frac{\pi}{\lambda}}\right)\left(\sqrt{\frac{\pi}{\lambda}}\right), \quad (51)$$

where  $\lambda = 1$ . Thus,  $F_1(k_x, k_y)$  can be expressed as

$$F_1(k_x, k_y) = I_0 e^{(i\pi k_x)^2} e^{(i\pi k_y)^2} \left(\sqrt{\frac{\pi}{1}}\right)\left(\sqrt{\frac{\pi}{1}}\right). \quad (52)$$

After simplifying it gives

$$F_1(k_x, k_y) = \pi I_0 e^{-\pi^2(k_x^2+k_y^2)}. \quad (53)$$

Next, one calculates the Fourier transform of  $f_2(x, y)$ ,  $F_2(k_x, k_y)$ , similar to the way for finding  $F_1(k_x, k_y)$ . The Fourier transform of  $f_2(x, y) = I_0 e^{-[(x-10)^2+(y-10)^2]}$ ,  $F_2(k_x, k_y)$  is

$$F_2(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_2(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (54)$$

$$F_2(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_0 e^{-[(x-10)^2+(y-10)^2]} e^{-i2\pi(k_x x + k_y y)} dx dy. \quad (55)$$

To evaluate this double integral analytically, one can make complete squares in the power of  $e$ , similar to the way for finding  $F_1(k_x, k_y)$ . Then, the equation above can be simplified as

$$F_2(k_x, k_y) = I_0 e^{-200} e^{(i\pi k_x - 10)^2} e^{(i\pi k_y - 10)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-[y+(i\pi k_y - 10)]^2} e^{-[x+(i\pi k_x - 10)]^2} dx dy. \quad (56)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}$  and  $\int_{-\infty}^{\infty} e^{-\lambda(y-c)^2} dy = \sqrt{\frac{\pi}{\lambda}}$  for any constant  $c$ , the integral

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-[y+(i\pi k_y-10)]^2} e^{-[x+(i\pi k_x-10)]^2} dx dy = \left(\sqrt{\frac{\pi}{\lambda}}\right)\left(\sqrt{\frac{\pi}{\lambda}}\right), \quad (57)$$

where  $\lambda = 1$ . Thus,  $F_2(k_x, k_y)$  can be expressed as

$$F_2(k_x, k_y) = I_0 e^{-200} e^{(i\pi k_x-10)^2} e^{(i\pi k_y-10)^2} \left(\sqrt{\frac{\pi}{1}}\right)\left(\sqrt{\frac{\pi}{1}}\right). \quad (58)$$

After simplifying it gives

$$F_2(k_x, k_y) = \pi I_0 e^{-200} e^{(i\pi k_x-10)^2} e^{(i\pi k_y-10)^2}. \quad (59)$$

Now taking complex conjugate of  $F_2(k_x, k_y)$  gives

$$F_2^*(k_x, k_y) = \pi I_0 e^{-200} e^{(-i\pi k_x-10)^2} e^{(-i\pi k_y-10)^2} \quad (60)$$

or

$$F_2^*(k_x, k_y) = \pi I_0 e^{-200} e^{(i\pi k_x+10)^2} e^{(i\pi k_y+10)^2}. \quad (61)$$

The cross-correlation function  $r_{x,y}$  is defined as

$$r_{x,y} = F^{-1} \left[ \frac{F_1(k_x, k_y) F_2^*(k_x, k_y)}{S_1 S_2} \right] \quad (62)$$

or

$$r_{x,y} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{F_1(k_x, k_y) F_2^*(k_x, k_y) e^{i2\pi(xk_x+yk_y)} dk_x dk_y}{S_1 S_2}, \quad (63)$$

where  $F_1(k_x, k_y)$  is the Fourier transform of  $f_1(x, y)$ ,  $F_2^*(k_x, k_y)$  the complex conjugate of the Fourier transform of  $f_2(x, y)$ ,  $S_1$  the standard deviation of  $f_1(x, y)$ ,

$S_2$  the standard deviation of  $f_2(x, y)$ , and  $F^{-1}$  represents the inverse Fourier transform. Substituting  $F_1(k_x, k_y) = \pi I_0 e^{-\pi^2(k_x^2 + k_y^2)}$  and

$F_2^*(k_x, k_y) = \pi I_0 e^{-200} e^{(i\pi k_x + 10)^2} e^{(i\pi k_y + 10)^2}$  into this equation, one has

$$r_{x,y} = \frac{1}{S_1 S_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [\pi I_0 e^{-\pi^2(k_x^2 + k_y^2)}] [\pi I_0 e^{-200} e^{(i\pi k_x + 10)^2} e^{(i\pi k_y + 10)^2}] e^{i2\pi(xk_x + yk_y)} dk_x dk_y. \quad (64)$$

As in 1-D case, one can simplify this expression by completing squares for the powers of e. After some algebra, the expression above becomes

$$r_{x,y} = \frac{\pi^2 I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{2}} e^{-\frac{(y+10)^2}{2}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-2\pi^2[k_y - \frac{i(y+10)}{2\pi}]^2} e^{-2\pi^2[k_x - \frac{i(x+10)}{2\pi}]^2} dk_x dk_y. \quad (65)$$

Now, let  $\lambda = 2\pi^2$ . Since  $\int_{-\infty}^{\infty} e^{-\lambda(k_x - c)^2} dk_x = \sqrt{\frac{\pi}{\lambda}}$  and  $\int_{-\infty}^{\infty} e^{-\lambda(k_y - c)^2} dk_y = \sqrt{\frac{\pi}{\lambda}}$  for any constant c, we have

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-2\pi^2[k_y - \frac{i(y+10)}{2\pi}]^2} e^{-2\pi^2[k_x - \frac{i(x+10)}{2\pi}]^2} dk_x dk_y = \left(\sqrt{\frac{\pi}{2\pi^2}}\right) \left(\sqrt{\frac{\pi}{2\pi^2}}\right) = \frac{1}{2\pi} \quad (66)$$

and the cross-correlation function  $r_{x,y}$  can be expressed as

$$r_{x,y} = \frac{\pi^2 I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{2}} e^{-\frac{(y+10)^2}{2}} \left(\frac{1}{2\pi}\right) \quad (67)$$

or

$$r_{x,y} = \frac{\pi I_0^2}{2S_1 S_2} e^{-\left[\frac{(x+10)^2 + (y+10)^2}{2}\right]}. \quad (68)$$

Finally, defining  $x_{lag} = -x$  and  $y_{lag} = -y$  for the lags, the cross-correlation function  $r_{x_{lag}, y_{lag}}$  can be expressed as

$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{2S_1 S_2} e^{-\left[\frac{(-x_{lag}+10)^2 + (-y_{lag}+10)^2}{2}\right]} \quad (69)$$

or

$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{2S_1 S_2} e^{-\left[\frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{2}\right]}, \quad (70)$$

where  $I_0$  is the peak backscatter intensity of the given Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ .

### One-Dimensional Diffused Gaussian Feature in an Uniform Flow

Suppose a 1-D Gaussian feature is moving at a constant rate in the positive x-direction and at the same time is diffused uniformly. In this case, it is also possible to find the cross-correlation function analytically. Let the 1-D Gaussian feature is at the origin at time  $t_1$ . Then, the backscatter image at time  $t_1$ ,  $f_1(x)$ , can be expressed as

$$f_1(x) = I_0 e^{-x^2}, \quad (71)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. Suppose further that the peak of the Gaussian feature moves 10 units to the right in the time interval  $\Delta t = t_2 - t_1$  and the Gaussian feature is diffused uniformly. Then, the backscatter image at time  $t_2$ ,  $f_2(x)$ , can be expressed as

$$f_2(x) = \frac{I_0}{\sqrt{2}} e^{-\frac{(x-10)^2}{2}}, \quad (72)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. The factor  $\frac{1}{2}$  in the exponential term represents the diffusion of the Gaussian feature and the

coefficient  $\frac{1}{\sqrt{2}}$  normalizes the Gaussian feature. Now, the 1-D cross-correlation function,  $r_x$ , applied to these images can be calculated as follows. The Fourier transform of  $f_1 = I_0 e^{-x^2}$ ,  $F_1(k)$  is

$$F_1(k) = \int_{-\infty}^{\infty} I_0 e^{-x^2} e^{-i2\pi kx} dx \quad (73)$$

or

$$F_1(k) = \int_{-\infty}^{\infty} I_0 e^{-(x^2 + i2\pi kx)} dx. \quad (74)$$

After evaluating the integral  $F_1(k)$  can be expressed as

$$F_1(k) = \sqrt{\pi} I_0 e^{-\pi^2 k^2} \quad (75)$$

as shown in previous section. Next, the Fourier transform of  $f_2(x) = \frac{I_0}{\sqrt{2}} e^{-\frac{(x-10)^2}{2}}$ ,  $F_2(k)$  is

$$F_2(k) = \int_{-\infty}^{\infty} \frac{I_0}{\sqrt{2}} e^{-\frac{(x-10)^2}{2}} e^{-i2\pi kx} dx. \quad (76)$$

As discussed previously, the same techniques can be applied to evaluate this integral. By completing squares, this integral becomes

$$F_2(k) = \frac{I_0}{\sqrt{2}} e^{-50} e^{\frac{1}{2}(i2\pi k-10)^2} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x+i2\pi k-10)^2} dx. \quad (77)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}$  for any constant  $c$ , the integral

$\int_{-\infty}^{\infty} e^{-\frac{1}{2}(x+i2\pi k-10)^2} dx = \sqrt{\frac{\pi}{\lambda}}$ , where  $\lambda = \frac{1}{2}$ . Thus,  $F_2(k)$  can be expressed as

$$F_2(k) = \frac{I_0}{\sqrt{2}} e^{-50} e^{\frac{1}{2}(i2\pi k-10)^2} \sqrt{\frac{\pi}{\frac{1}{2}}} \quad (78)$$

or

$$F_2(k) = \sqrt{\pi}I_0e^{-50}e^{\frac{1}{2}(i2\pi k-10)^2}. \quad (79)$$

Now taking complex conjugate of  $F_2(k)$  gives

$$F_2^*(k) = \sqrt{\pi}I_0e^{-50}e^{\frac{1}{2}(-i2\pi k-10)^2} \quad (80)$$

or

$$F_2^*(k) = \sqrt{\pi}I_0e^{-50}e^{\frac{1}{2}(i2\pi k+10)^2}. \quad (81)$$

The cross-correlation function  $r_x$  is defined as

$$r_x = F^{-1} \left[ \frac{F_1(k)F_2^*(k)}{S_1S_2} \right] \quad (82)$$

or

$$r_x = \int_{-\infty}^{\infty} \frac{F_1(k)F_2^*(k)e^{i2\pi kx}}{S_1S_2} dk, \quad (83)$$

where  $F_1(k)$  is the Fourier transform of  $f_1(x)$ ,  $F_2^*(k)$  the complex conjugate of the Fourier transform of  $f_2(x)$ ,  $S_1$  the standard deviation of  $f_1(x)$ ,  $S_2$  the standard deviation of  $f_2(x)$ , and  $F^{-1}$  represents the inverse Fourier transform. Substituting  $F_1(k) = \sqrt{\pi}I_0e^{-\pi^2k^2}$  and  $F_2^*(k) = \sqrt{\pi}I_0e^{-50}e^{\frac{1}{2}(i2\pi k+10)^2}$  into this equation, one has

$$r_x = \frac{1}{S_1S_2} \int_{-\infty}^{\infty} \sqrt{\pi}I_0e^{-\pi^2k^2} \sqrt{\pi}I_0e^{-50}e^{\frac{1}{2}(i2\pi k+10)^2} e^{i2\pi kx} dk \quad (84)$$

$$r_x = \frac{\pi I_0^2}{S_1S_2} e^{-50} \int_{-\infty}^{\infty} e^{-\pi^2k^2} e^{\frac{1}{2}(i2\pi k+10)^2} e^{i2\pi kx} dk. \quad (85)$$

As in previous section, we can simplify  $r_x$ . After some algebra  $r_x$  becomes

$$r_x = \frac{\pi I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{3}} \int_{-\infty}^{\infty} e^{-3\pi^2 [k - \frac{i(x+10)}{3\pi}]^2} dk. \quad (86)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(k-c)^2} dk = \sqrt{\frac{\pi}{\lambda}}$  for any constant  $c$ , the integral  $\int_{-\infty}^{\infty} e^{-3\pi^2 [k - \frac{i(x+10)}{3\pi}]^2} dk = \sqrt{\frac{\pi}{\lambda}}$ , where  $\lambda = 3\pi^2$ . Thus,

$$r_x = \frac{\pi I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{3}} \sqrt{\frac{\pi}{3\pi^2}}. \quad (87)$$

After simplifying it gives

$$r_x = \sqrt{\frac{\pi}{3}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{3}}. \quad (88)$$

Finally, defining  $x_{lag} = -x$  for the lag, the cross-correlation function  $r_{x_{lag}}$  can be expressed as

$$r_{x_{lag}} = \sqrt{\frac{\pi}{3}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(-x_{lag}+10)^2}{3}} \quad (89)$$

or

$$r_{x_{lag}} = \sqrt{\frac{\pi}{3}} \frac{I_0^2}{S_1 S_2} e^{-\frac{(x_{lag}-10)^2}{3}}, \quad (90)$$

where  $I_0$  is the peak backscatter intensity of the given Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x)$ , and  $S_2$  is the standard deviation of  $f_2(x)$ .

### Two-Dimensional Diffused Gaussian Feature in an Uniform Flow

One can extend the concepts of the 1-D diffused Gaussian feature, described above, to consider the 2-D case. Suppose that there is only one Gaussian feature in a 2-D space, and the Gaussian feature is at the origin,  $(x, y) = (0, 0)$ , at time  $t_1$ . Then,

the backscatter image at time  $t_1$ ,  $f_1(x, y)$ , can be expressed as

$$f_1(x, y) = I_0 e^{-(x^2+y^2)}, \quad (91)$$

where  $I_0$  is the peak backscatter image of the Gaussian feature. Suppose further that the Gaussian feature moves 10 units to the right (positive x direction), 10 units to the upward (positive y direction), and diffuses uniformly in the xy-plane in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image at time  $t_2$ ,  $f_2(x, y)$ , can be expressed as

$$f_2(x, y) = \frac{I_0}{2} e^{-\frac{[(x-10)^2+(y-10)^2]}{2}}, \quad (92)$$

where  $I_0$  is the peak backscatter intensity of the Gaussian feature. The 2-D cross-correlation function,  $r_{x,y}$ , applied to these images can be calculated as follows. The Fourier transform of  $f_1(x, y) = I_0 e^{-(x^2+y^2)}$ ,  $F_1(k_x, k_y)$  is

$$F_1(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (93)$$

or

$$F_1(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_0 e^{-(x^2+y^2)} e^{-i2\pi(k_x x + k_y y)} dx dy. \quad (94)$$

After evaluating the integral  $F_1(k_x, k_y)$  can be expressed as

$$F_1(k_x, k_y) = \pi I_0 e^{-\pi^2(k_x^2+k_y^2)} \quad (95)$$

as shown in previous section. Next, the Fourier transform of

$$f_2(x, y) = \frac{I_0}{2} e^{-\frac{[(x-10)^2+(y-10)^2]}{2}}, F_2(k_x, k_y) \text{ is}$$

$$F_2(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_2(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (96)$$

or

$$F_2(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{I_0}{2} e^{-\frac{[(x-10)^2+(y-10)^2]}{2}} e^{-i2\pi(k_x x + k_y y)} dx dy. \quad (97)$$

To evaluate this double integral analytically, one can make complete squares in the power of e, as we did in the previous section. Then, the equation above can be simplified as

$$F_2(k_x, k_y) = \frac{I_0}{2} e^{-100} e^{\frac{1}{2}(i2\pi k_x - 10)^2} e^{\frac{1}{2}(i2\pi k_y - 10)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{1}{2}[y+(i2\pi k_y - 10)]^2} e^{-\frac{1}{2}[x+(i2\pi k_x - 10)]^2} dx dy. \quad (98)$$

Since  $\int_{-\infty}^{\infty} e^{-\lambda(x-c)^2} dx = \sqrt{\frac{\pi}{\lambda}}$  and  $\int_{-\infty}^{\infty} e^{-\lambda(y-c)^2} dy = \sqrt{\frac{\pi}{\lambda}}$  for any constant c, the integral becomes

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{1}{2}[y+(i2\pi k_y - 10)]^2} e^{-\frac{1}{2}[x+(i2\pi k_x - 10)]^2} dx dy = \left(\sqrt{\frac{\pi}{\lambda}}\right) \left(\sqrt{\frac{\pi}{\lambda}}\right), \quad (99)$$

where  $\lambda = \frac{1}{2}$ . Thus,  $F_2(k_x, k_y)$  can be expressed as

$$F_2(k_x, k_y) = \frac{I_0}{2} e^{-100} e^{\frac{1}{2}(i2\pi k_x - 10)^2} e^{\frac{1}{2}(i2\pi k_y - 10)^2} \sqrt{\frac{\pi}{(\frac{1}{2})}} \sqrt{\frac{\pi}{(\frac{1}{2})}} \quad (100)$$

or, by simplifying it gives

$$F_2(k_x, k_y) = \pi I_0 e^{-100} e^{\frac{1}{2}(i2\pi k_x - 10)^2} e^{\frac{1}{2}(i2\pi k_y - 10)^2}. \quad (101)$$

Now taking complex conjugate of  $F_2(k_x, k_y)$  gives

$$F_2^*(k_x, k_y) = \pi I_0 e^{-100} e^{\frac{1}{2}(-i2\pi k_x - 10)^2} e^{\frac{1}{2}(-i2\pi k_y - 10)^2} \quad (102)$$

or

$$F_2^*(k_x, k_y) = \pi I_0 e^{-100} e^{\frac{1}{2}(i2\pi k_x + 10)^2} e^{\frac{1}{2}(i2\pi k_y + 10)^2}. \quad (103)$$

The cross-correlation function  $r_{x,y}$  is defined as

$$r_{x,y} = F^{-1} \left[ \frac{F_1(k_x, k_y) F_2^*(k_x, k_y)}{S_1 S_2} \right] \quad (104)$$

or

$$r_{x,y} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{F_1(k_x, k_y) F_2^*(k_x, k_y) e^{i2\pi(xk_x + yk_y)} dk_x dk_y}{S_1 S_2}, \quad (105)$$

where  $F_1(k_x, k_y)$  is the Fourier transform of  $f_1(x, y)$ ,  $F_2^*(k_x, k_y)$  the complex conjugate of the Fourier transform of  $f_2(x, y)$ ,  $S_1$  the standard deviation of  $f_1(x, y)$ ,  $S_2$  the standard deviation of  $f_2(x, y)$ , and  $F^{-1}$  represents the inverse Fourier transform. Substituting  $F_1(k_x, k_y) = \pi I_0 e^{-\pi^2(k_x^2 + k_y^2)}$  and

$F_2^*(k_x, k_y) = \pi I_0 e^{-100} e^{\frac{1}{2}(i2\pi k_x + 10)^2} e^{\frac{1}{2}(i2\pi k_y + 10)^2}$  into this equation, one has

$$r_{x,y} = \frac{1}{S_1 S_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [\pi I_0 e^{-\pi^2(k_x^2 + k_y^2)}] [\pi I_0 e^{-100} e^{\frac{1}{2}(i2\pi k_x + 10)^2} e^{\frac{1}{2}(i2\pi k_y + 10)^2}] e^{i2\pi(xk_x + yk_y)} dk_x dk_y. \quad (106)$$

As in the 1-D diffused Gaussian feature case, one can simplify this expression by completing squares for the powers of e. After some algebra, the expression above becomes

$$r_{x,y} = \frac{\pi^2 I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{3}} e^{-\frac{(y+10)^2}{3}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-3\pi^2 [k_y - \frac{i(y-10)}{3\pi}]^2} e^{-3\pi^2 [k_x - \frac{i(x-10)}{3\pi}]^2} dk_x dk_y. \quad (107)$$

Now, let  $\lambda = 3\pi^2$ . Since  $\int_{-\infty}^{\infty} e^{-\lambda(k_x-c)^2} dk_x = \sqrt{\frac{\pi}{\lambda}}$  and  $\int_{-\infty}^{\infty} e^{-\lambda(k_y-c)^2} dk_y = \sqrt{\frac{\pi}{\lambda}}$  for any constant  $c$ , one has

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-3\pi^2[k_y - \frac{i(y-10)}{3\pi}]^2} e^{-3\pi^2[k_x - \frac{i(x-10)}{3\pi}]^2} dk_x dk_y = \left(\sqrt{\frac{\pi}{3\pi^2}}\right)\left(\sqrt{\frac{\pi}{3\pi^2}}\right) = \frac{1}{3\pi} \quad (108)$$

and the cross-correlation function  $r_{x,y}$  can be expressed as

$$r_{x,y} = \frac{\pi^2 I_0^2}{S_1 S_2} e^{-\frac{(x+10)^2}{3}} e^{-\frac{(y+10)^2}{3}} \left(\frac{1}{3\pi}\right) \quad (109)$$

or

$$r_{x,y} = \frac{\pi I_0^2}{3S_1 S_2} e^{-\left[\frac{(x+10)^2 + (y+10)^2}{3}\right]}. \quad (110)$$

Finally, defining  $x_{lag} = -x$  and  $y_{lag} = -y$  for the lags, the cross-correlation function  $r_{x_{lag}, y_{lag}}$  can be expressed as

$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{3S_1 S_2} e^{-\left[\frac{(-x_{lag}+10)^2 + (-y_{lag}+10)^2}{3}\right]} \quad (111)$$

or

$$r_{x_{lag}, y_{lag}} = \frac{\pi I_0^2}{3S_1 S_2} e^{-\left[\frac{(x_{lag}-10)^2 + (y_{lag}-10)^2}{3}\right]}, \quad (112)$$

where  $I_0$  is the peak backscatter intensity of the given Gaussian feature,  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ .

## APPENDIX B

## APPENDIX B

### THE CROSS-CORRELATION FUNCTION FOR A RECTANGULAR PULSE

#### Analytical Approaches Using a Rectangular Pulse

##### Contour Integration

To calculate the cross-correlation function for a rectangular pulse, one needs to evaluate the following integral.

$$I = \int_0^{\infty} \frac{\sin(\lambda x)}{x} dx, \quad (1)$$

where  $\lambda$  is a positive constant. To evaluate this integral, one can apply "Calculus of Residues" described by (Arfken and Weber, 2001) and (Mathews and Walker, 1970). Let  $I_0 = \oint \frac{e^{i\lambda z}}{z} dz$  where  $z = x + iy$  is a complex number, and take a contour as shown as Figure 69. The contour of the integral  $I_0 = \oint \frac{e^{i\lambda z}}{z} dz$  can be divided by 4 sections  $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_4$  in Figure 69.

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = \int_{C_1} \frac{e^{i\lambda z}}{z} dz + \int_{C_2} \frac{e^{i\lambda z}}{z} dz + \int_{C_3} \frac{e^{i\lambda z}}{z} dz + \int_{C_4} \frac{e^{i\lambda z}}{z} dz. \quad (2)$$

From Figure 69, one can see that  $C_1$  is on the real axis. Then,  $z = x$  and  $-R < x < -r$ , and one has

$$\int_{C_1} \frac{e^{i\lambda z}}{z} dz = \int_{-R}^{-r} \frac{e^{i\lambda x}}{x} dx. \quad (3)$$

For the path  $C_2$ , let  $z = re^{i\phi}$  and  $dz = ire^{i\phi}d\phi$ , where  $\phi$  changes from  $\pi$  to 0 and  $r$  approaches 0. Then, one has

$$\int_{C_2} \frac{e^{i\lambda z}}{z} dz = \int_{\pi}^0 \frac{e^{i\lambda z}}{(re^{i\phi})} (ire^{i\phi} d\phi) \quad (4)$$

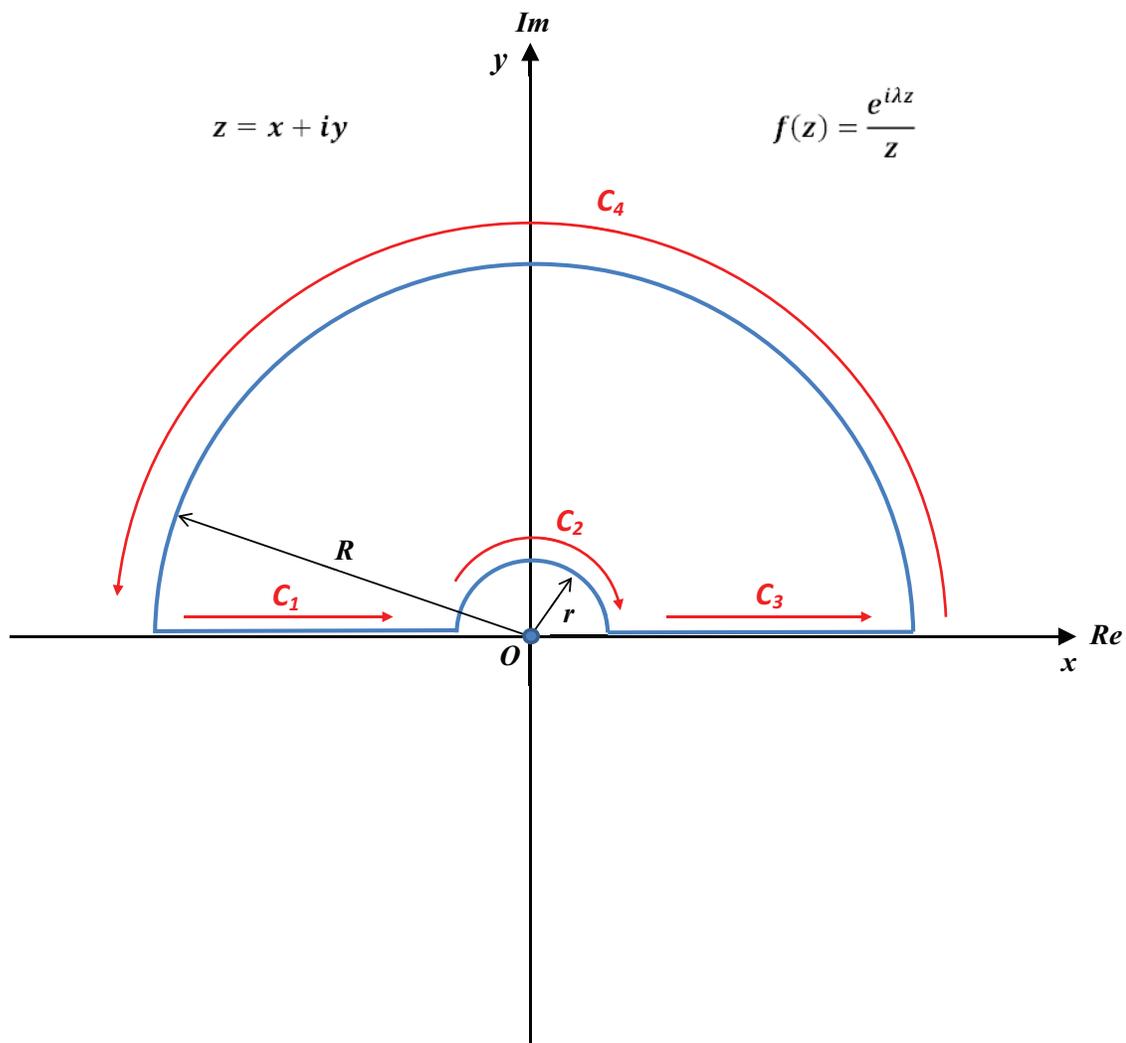


Figure 69. Contour integration of  $I_0 = \oint f(z) dz$  where  $f(z) = \frac{e^{i\lambda z}}{z}$  and  $z = x + iy$  is a complex number.

or it is reduced to

$$\int_{C_2} \frac{e^{i\lambda z}}{z} dz = i \int_{\pi}^0 e^{i\lambda z} d\phi. \quad (5)$$

In the case of  $r$  approaching 0,  $z = re^{i\phi}$  also approaches zero. Thus,  $e^{i\lambda z} = e^0 = 1$ , and one has

$$\int_{C_2} \frac{e^{i\lambda z}}{z} dz = i \int_{\pi}^0 1 d\phi = i(0 - \pi) = -i\pi. \quad (6)$$

As the path  $C_1$ , one can see that  $C_3$  is also on the real axis. Then,  $z = x$  and  $r < x < R$ , and one has

$$\int_{C_3} \frac{e^{i\lambda z}}{z} dz = \int_r^R \frac{e^{i\lambda x}}{x} dx. \quad (7)$$

The integral along the path  $C_4$  is zero, according to Jordan's lemma as  $R$  approaches infinity.

$$\int_{C_4} \frac{e^{i\lambda z}}{z} dz = 0. \quad (8)$$

Thus the contour integral along the given path becomes

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = \int_{-R}^{-r} \frac{e^{i\lambda x}}{x} dx - i\pi + \int_r^R \frac{e^{i\lambda x}}{x} dx + 0. \quad (9)$$

As  $r$  approaches 0 and  $R$  approaches infinity, the integral becomes

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = \int_{-\infty}^0 \frac{e^{i\lambda x}}{x} dx - i\pi + \int_0^{\infty} \frac{e^{i\lambda x}}{x} dx + 0, \quad (10)$$

which is reduced to

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = \int_{-\infty}^{\infty} \frac{e^{i\lambda x}}{x} dx - i\pi. \quad (11)$$

From the Euler's formula  $e^{i\lambda x} = \cos(\lambda x) + i \sin(\lambda x)$ , so one can write

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = \int_{-\infty}^{\infty} \frac{[\cos(\lambda x) + i \sin(\lambda x)]}{x} dx - i\pi \quad (12)$$

or separating real and imaginary parts gives

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = \int_{-\infty}^{\infty} \frac{\cos(\lambda x)}{x} dx + i \left( \int_{-\infty}^{\infty} \frac{\sin(\lambda x)}{x} dx - \pi \right). \quad (13)$$

Now, from the Residue Theorem, one has

$$\oint f(z) dz = 2\pi i \sum(\text{Residues}). \quad (14)$$

From Figure 69, one can see that no poles (singularities) are enclosed in the given contour. Thus,  $\sum(\text{Residues}) = 0$ , for the function  $f(z) = \frac{e^{i\lambda z}}{z}$  and one has

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = 0. \quad (15)$$

By comparing the real and the imaginary parts of two equations

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = \int_{-\infty}^{\infty} \frac{\cos(\lambda x)}{x} dx + i \left( \int_{-\infty}^{\infty} \frac{\sin(\lambda x)}{x} dx - \pi \right) \quad (16)$$

and

$$I_0 = \oint \frac{e^{i\lambda z}}{z} dz = 0 \quad (17)$$

one can get

$$\int_{-\infty}^{\infty} \frac{\cos(\lambda x)}{x} dx = 0 \quad (18)$$

and

$$\int_{-\infty}^{\infty} \frac{\sin(\lambda x)}{x} dx - \pi = 0 \quad (19)$$

$$\int_{-\infty}^{\infty} \frac{\sin(\lambda x)}{x} dx = \pi. \quad (20)$$

Now, both  $\sin(\lambda x)$  and  $x$  are odd functions, in which case the function  $\frac{\sin(\lambda x)}{x}$  must be an even function. Thus,

$$\int_0^{\infty} \frac{\sin(\lambda x)}{x} dx = \frac{1}{2} \int_{-\infty}^{\infty} \frac{\sin(\lambda x)}{x} dx. \quad (21)$$

That is,

$$I = \int_0^{\infty} \frac{\sin(\lambda x)}{x} dx = \frac{\pi}{2}. \quad (22)$$

This equation is helpful to evaluate the cross-correlation function for rectangular pulses moving at constant velocity.

#### One-Dimensional Rectangular Pulse in an Uniform Flow

Suppose that there is only 1 rectangular pulse in a 1-D space, and the center of the rectangular pulse is at the origin,  $x = 0$ , at time  $t_1$ . Then, the backscatter image at time  $t_1$ ,  $f_1(x)$ , can be expressed as

$$f_1(x) = \begin{cases} I_0 & : -\frac{a}{2} \leq x \leq \frac{a}{2} \\ 0 & : otherwise \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the rectangular pulse. Suppose further that the rectangular pulse moves 10 units to the right in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image at time  $t_2$ ,  $f_2(x)$ , can be expressed as

$$f_2(x) = \begin{cases} I_0 & : -\frac{a}{2} + 10 \leq x \leq \frac{a}{2} + 10 \\ 0 & : otherwise \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the rectangular pulse. Now, the 1-D cross-correlation function,  $r_x$ , applied to these images can be calculated as follows.

The Fourier transform of  $f_1(x)$ ,  $F_1(k)$  is

$$F_1(k) = \int_{-\infty}^{\infty} f_1(x)e^{-i2\pi kx} dx \quad (23)$$

$$F_1(k) = \int_{-\frac{a}{2}}^{\frac{a}{2}} I_0 e^{-i2\pi kx} dx. \quad (24)$$

To evaluate this integral, let

$$u = -i2\pi kx. \quad (25)$$

Then

$$du = -i2\pi k dx \quad (26)$$

or

$$-\frac{1}{i2\pi k} du = dx. \quad (27)$$

Using these expressions, the given integral  $F_1(k) = \int_{-\frac{a}{2}}^{\frac{a}{2}} I_0 e^{-i2\pi kx} dx$  becomes

$$F_1(k) = I_0 \int_{x=-\frac{a}{2}}^{x=\frac{a}{2}} (e^u) \left( -\frac{1}{i2\pi k} \right) du \quad (28)$$

$$F_1(k) = -\frac{I_0}{i2\pi k} e^u \Big|_{x=-\frac{a}{2}}^{x=\frac{a}{2}} \quad (29)$$

or, back substituting  $u = -i2\pi kx$ , one has

$$F_1(k) = -\frac{I_0}{i2\pi k} e^{-i2\pi kx} \Big|_{x=-\frac{a}{2}}^{x=\frac{a}{2}} \quad (30)$$

$$F_1(k) = -\frac{I_0}{i2\pi k} (e^{-ia\pi k} - e^{ia\pi k}) \quad (31)$$

$$F_1(k) = \left( \frac{I_0}{\pi k} \right) \left( \frac{e^{ia\pi k} - e^{-ia\pi k}}{2i} \right). \quad (32)$$

Now,  $\sin(a\pi k) = \frac{e^{ia\pi k} - e^{-ia\pi k}}{2i}$ . Thus,  $F_1(k)$  can be expressed as

$$F_1(k) = \left( \frac{I_0}{\pi k} \right) \sin(a\pi k). \quad (33)$$

Next, the Fourier transform of  $f_2(x)$ ,  $F_2(k)$  is

$$F_2(k) = \int_{-\infty}^{\infty} f_2(x) e^{-i2\pi kx} dx \quad (34)$$

$$F_2(k) = \int_{-\frac{a}{2}+10}^{\frac{a}{2}+10} I_0 e^{-i2\pi kx} dx. \quad (35)$$

As the case of evaluating  $F_1(k)$ , let

$$u = -i2\pi kx \quad (36)$$

and

$$-\frac{1}{i2\pi k} du = dx. \quad (37)$$

Then, the given integral  $F_2(k) = \int_{-\frac{a}{2}+10}^{\frac{a}{2}+10} I_0 e^{-i2\pi kx} dx$  becomes

$$F_2(k) = I_0 \int_{x=-\frac{a}{2}+10}^{x=\frac{a}{2}+10} (e^u) \left( -\frac{1}{i2\pi k} \right) du \quad (38)$$

$$F_2(k) = -\frac{I_0}{i2\pi k} e^u \Big|_{x=-\frac{a}{2}+10}^{x=\frac{a}{2}+10} \quad (39)$$

or, back substituting  $u = -i2\pi kx$ , one has

$$F_2(k) = -\frac{I_0}{i2\pi k} e^{-i2\pi kx} \Big|_{x=-\frac{a}{2}+10}^{x=\frac{a}{2}+10} \quad (40)$$

$$F_2(k) = -\frac{I_0}{i2\pi k} [e^{-i2\pi k(\frac{a}{2}+10)} - e^{-i2\pi k(-\frac{a}{2}+10)}] \quad (41)$$

which can be simplified as

$$F_2(k) = \left( \frac{I_0 e^{-i20\pi k}}{\pi k} \right) \left( \frac{e^{ia\pi k} - e^{-ia\pi k}}{2i} \right). \quad (42)$$

Now,  $\sin(a\pi k) = \frac{e^{ia\pi k} - e^{-ia\pi k}}{2i}$ . Thus,  $F_2(k)$  can be expressed as

$$F_2(k) = \left( \frac{I_0 e^{-i20\pi k}}{\pi k} \right) \sin(a\pi k). \quad (43)$$

Now taking complex conjugate of  $F_2(k)$  gives

$$F_2^*(k) = \left( \frac{I_0 e^{i20\pi k}}{\pi k} \right) \sin(a\pi k). \quad (44)$$

The cross-correlation function  $r_x$  is defined as

$$r_x = F^{-1} \left[ \frac{F_1(k) F_2^*(k)}{S_1 S_2} \right] \quad (45)$$

or

$$r_x = \int_{-\infty}^{\infty} \frac{F_1(k) F_2^*(k) e^{i2\pi kx}}{S_1 S_2} dk, \quad (46)$$

where  $F_1(k)$  is the Fourier transform of  $f_1(x)$ ,  $F_2^*(k)$  the complex conjugate of the Fourier transform of  $f_2(x)$ ,  $S_1$  the standard deviation of  $f_1(x)$ ,  $S_2$  the standard deviation of  $f_2(x)$ , and  $F^{-1}$  represents the inverse Fourier transform. Substituting  $F_1(k) = \left( \frac{I_0}{\pi k} \right) \sin(a\pi k)$  and  $F_2^*(k) = \left( \frac{I_0 e^{i20\pi k}}{\pi k} \right) \sin(a\pi k)$  into the expression of  $r_x$ , one has

$$r_x = \frac{1}{S_1 S_2} \int_{-\infty}^{\infty} \left( \frac{I_0}{\pi k} \right) \sin(a\pi k) \left( \frac{I_0 e^{i20\pi k}}{\pi k} \right) \sin(a\pi k) e^{i2\pi kx} dk \quad (47)$$

or it is reduced to

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_{-\infty}^{\infty} \frac{e^{i2\pi(x+10)k} \sin^2(a\pi k)}{k^2} dk. \quad (48)$$

Using the Euler's formula,  $e^{i2\pi(x+10)k} = \cos[2\pi(x+10)k] + i \sin[2\pi(x+10)k]$ , one has

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_{-\infty}^{\infty} \frac{(\cos[2\pi(x+10)k] + i \sin[2\pi(x+10)k]) \sin^2(a\pi k)}{k^2} dk \quad (49)$$

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_{-\infty}^{\infty} \frac{\cos[2\pi(x+10)k] \sin^2(a\pi k)}{k^2} dk + i \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_{-\infty}^{\infty} \frac{\sin[2\pi(x+10)k] \sin^2(a\pi k)}{k^2} dk. \quad (50)$$

Now,  $\frac{\sin[2\pi(x+10)k] \sin^2(a\pi k)}{k^2}$  is an odd function. Then,

$$\int_{-\infty}^{\infty} \frac{\sin[2\pi(x+10)k] \sin^2(a\pi k)}{k^2} dk = 0. \quad (51)$$

On the other hand,  $\frac{\cos[2\pi(x+10)k] \sin^2(a\pi k)}{k^2}$  is an even function. Thus, one has

$$\int_{-\infty}^{\infty} \frac{\cos[2\pi(x+10)k] \sin^2(a\pi k)}{k^2} dk = 2 \int_0^{\infty} \frac{\cos[2\pi(x+10)k] \sin^2(a\pi k)}{k^2} dk. \quad (52)$$

Using these expressions, the cross-correlation function  $r_x$  can be expressed as

$$r_x = 2 \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\cos[2\pi(x+10)k] \sin^2(a\pi k)}{k^2} dk. \quad (53)$$

To evaluate this integral, let  $\alpha = a\pi$  and  $\beta = \pi(x + 10)$ . Then, one has

$$r_x = 2 \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\cos(2\beta k) \sin^2(\alpha k)}{k^2} dk. \quad (54)$$

Next, applying the trigonometric identity,  $\sin^2(\alpha k) = \frac{1}{2}[1 - \cos(2\alpha k)]$ , one has

$$r_x = 2 \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\cos(2\beta k) \frac{1}{2}[1 - \cos(2\alpha k)]}{k^2} dk \quad (55)$$

which is,

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\cos(2\beta k) - \cos(2\beta k) \cos(2\alpha k)}{k^2} dk \quad (56)$$

or

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\cos(2\beta k) - \cos(2\alpha k) \cos(2\beta k)}{k^2} dk. \quad (57)$$

Now, the term  $\cos(2\alpha k) \cos(2\beta k)$  can be expressed as

$$\cos(2\alpha k) \cos(2\beta k) = \frac{1}{2} \cos[2(\alpha - \beta)k] + \frac{1}{2} \cos[2(\alpha + \beta)k]. \quad (58)$$

Using this expression, the cross-correlation function  $r_x$  can be expressed as

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\cos(2\beta k) - \left\{ \frac{1}{2} \cos[2(\alpha - \beta)k] + \frac{1}{2} \cos[2(\alpha + \beta)k] \right\}}{k^2} dk \quad (59)$$

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\cos(2\beta k) - \frac{1}{2} \cos[2(\alpha - \beta)k] - \frac{1}{2} \cos[2(\alpha + \beta)k]}{k^2} dk. \quad (60)$$

Next, let

$$u = \cos(2\beta k) - \frac{1}{2} \cos[2(\alpha - \beta)k] - \frac{1}{2} \cos[2(\alpha + \beta)k] \quad (61)$$

$$du = \{-2\beta \sin(2\beta k) + (\alpha - \beta) \sin[2(\alpha - \beta)k] + (\alpha + \beta) \sin[2(\alpha + \beta)k]\} dk \quad (62)$$

$$dv = \frac{1}{k^2} dk \quad (63)$$

and

$$v = -\frac{1}{k}. \quad (64)$$

Then, the cross-correlation  $r_x$  can be expressed as

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_{k=0}^{k \rightarrow \infty} u dv. \quad (65)$$

Now, the Integration by Parts gives

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \left\{ uv \Big|_{k=0}^{k \rightarrow \infty} - \int_{k=0}^{k \rightarrow \infty} v du \right\} \quad (66)$$

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \left\{ \left[ -\frac{\cos(2\beta k) - \frac{1}{2} \cos[2(\alpha - \beta)k] - \frac{1}{2} \cos[2(\alpha + \beta)k]}{k} \right] \Big|_{k=0}^{k \rightarrow \infty} - \int_{k=0}^{k \rightarrow \infty} v du \right\}. \quad (67)$$

By applying the L' Hopital's rule, the term,

$$\left[ -\frac{\cos(2\beta k) - \frac{1}{2} \cos[2(\alpha - \beta)k] - \frac{1}{2} \cos[2(\alpha + \beta)k]}{k} \right] \Big|_{k=0}^{k \rightarrow \infty} = 0. \text{ Thus one has}$$

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \left\{ 0 - \int_{k=0}^{k \rightarrow \infty} v du \right\} \quad (68)$$

$$r_x = - \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \left( -\frac{1}{k} \right) \{-2\beta \sin(2\beta k) + (\alpha - \beta) \sin[2(\alpha - \beta)k] + (\alpha + \beta) \sin[2(\alpha + \beta)k]\} dk \quad (69)$$

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \int_0^{\infty} \frac{\{-2\beta \sin(2\beta k) + (\alpha - \beta) \sin[2(\alpha - \beta)k] + (\alpha + \beta) \sin[2(\alpha + \beta)k]\}}{k} dk \quad (70)$$

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \left\{ -2\beta \int_0^{\infty} \frac{\sin(2\beta k)}{k} dk + (\alpha - \beta) \int_0^{\infty} \frac{\sin[2(\alpha - \beta)k]}{k} dk + (\alpha + \beta) \int_0^{\infty} \frac{\sin[2(\alpha + \beta)k]}{k} dk \right\}. \quad (71)$$

From the previous section, one found that

$$\int_0^{\infty} \frac{\sin(\lambda x)}{x} dx = \frac{\pi}{2}, \quad (72)$$

where  $\lambda$  is a positive constant. Then, one has

$$\int_0^{\infty} \frac{\sin(2\beta k)}{k} dk = \frac{\pi}{2} \quad (73)$$

$$\int_0^{\infty} \frac{\sin[2(\alpha - \beta)k]}{k} dk = \frac{\pi}{2} \quad (74)$$

and

$$\int_0^{\infty} \frac{\sin[2(\alpha + \beta)k]}{k} dk = \frac{\pi}{2}, \quad (75)$$

where  $\beta > 0$ ,  $\alpha - \beta > 0$  and  $\alpha + \beta > 0$ . Using these expressions, the cross-correlation function  $r_x$  can be expressed as

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \left[ -2\beta \left( \frac{\pi}{2} \right) + (\alpha - \beta) \left( \frac{\pi}{2} \right) + (\alpha + \beta) \left( \frac{\pi}{2} \right) \right] \quad (76)$$

$$r_x = \left( \frac{I_0^2}{\pi^2 S_1 S_2} \right) \left[ \pi \left( -\beta + \frac{(\alpha - \beta)}{2} + \frac{(\alpha + \beta)}{2} \right) \right] \quad (77)$$

which is reduced to

$$r_x = \left( \frac{I_0^2}{\pi S_1 S_2} \right) (\alpha - \beta). \quad (78)$$

For  $x > -10$ ,  $\beta = \pi(x + 10)$  and  $\alpha = a\pi$ . Thus the cross-correlation function  $r_x$  becomes

$$r_x = \left( \frac{I_0^2}{\pi S_1 S_2} \right) [a\pi - \pi(x + 10)] \quad (79)$$

$$r_x = \left( \frac{I_0^2}{S_1 S_2} \right) (a - x - 10). \quad (80)$$

As Gaussian features cases as shown in Appendix A, let  $x_{lag} = -x$ . Then the cross-correlation function  $r_{x_{lag}}$  becomes

$$r_{x_{lag}} = \left( \frac{I_0^2}{S_1 S_2} \right) (a + x_{lag} - 10). \quad (81)$$

That is

$$r_{x_{lag}} = \left( \frac{I_0^2}{S_1 S_2} \right) a + \left( \frac{I_0^2}{S_1 S_2} \right) (x_{lag} - 10). \quad (82)$$

This expression is valid when  $x > -10$ , that is  $x_{lag} < 10$ . Since,  $r_x = 0$  when  $x_{lag} = 10 - a$ , the range of  $x_{lag}$  must be  $10 - a < x_{lag} < 10$ . For  $x < -10$ ,  $\beta = -\pi(x + 10)$  and  $\alpha = a\pi$ . Thus the cross-correlation function  $r_x$  becomes

$$r_x = \left( \frac{I_0^2}{\pi S_1 S_2} \right) \{a\pi - [-\pi(x + 10)]\} \quad (83)$$

$$r_x = \left( \frac{I_0^2}{S_1 S_2} \right) (a + x + 10). \quad (84)$$

Let  $x_{lag} = -x$ . Then the cross-correlation function  $r_{x_{lag}}$  becomes

$$r_{x_{lag}} = \left( \frac{I_0^2}{S_1 S_2} \right) (a - x_{lag} + 10). \quad (85)$$

That is

$$r_{x_{lag}} = \left( \frac{I_0^2}{S_1 S_2} \right) a - \left( \frac{I_0^2}{S_1 S_2} \right) (x_{lag} - 10). \quad (86)$$

This expression is valid when  $x < -10$ , that is  $x_{lag} > 10$ . Since,  $r_{x_{lag}} = 0$  when  $x_{lag} = 10 + a$ , the range of  $x_{lag}$  must be  $10 < x_{lag} < 10 + a$ . Finally, one combines

these results and the cross-correlation function  $r_{x_{lag}}$  can be expressed as

$$r_{x_{lag}} = \begin{cases} \left( \frac{I_0^2}{S_1 S_2} \right) a + \left( \frac{I_0^2}{S_1 S_2} \right) (x_{lag} - 10) & : 10 - a \leq x_{lag} \leq 10 \\ \left( \frac{I_0^2}{S_1 S_2} \right) a - \left( \frac{I_0^2}{S_1 S_2} \right) (x_{lag} - 10) & : 10 \leq x_{lag} \leq 10 + a \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the given rectangular pulse,  $a$  is the width of the rectangular pulse,  $S_1$  is the standard deviation of  $f_1(x)$ , and  $S_2$  is the standard deviation of  $f_2(x)$ .

### Two-Dimensional Rectangular Pulse in an Uniform Flow

One can extend the concepts of the 1-D case, described above, to consider the 2-D case. Suppose that there is only 1 rectangular pulse in a 2-D space, and the center of the rectangular pulse is at the origin,  $(x, y) = (0, 0)$ , at time  $t_1$ . Then, the backscatter image at time  $t_1$ ,  $f_1(x, y)$ , can be expressed as

$$f_1(x, y) = \begin{cases} I_0 & : -\frac{a}{2} \leq x \leq \frac{a}{2} : -\frac{b}{2} \leq y \leq \frac{b}{2} \\ 0 & : otherwise \end{cases}$$

where  $I_0$  is the peak backscatter image of the rectangular pulse. Suppose further that the rectangular pulse moves 10 units to the right (positive x direction) and 10 units to the upward (positive y direction) in the time interval  $\Delta t = t_2 - t_1$ . Then, the backscatter image at time  $t_2$ ,  $f_2(x, y)$ , can be expressed as

$$f_2(x, y) = \begin{cases} I_0 & : 10 - \frac{a}{2} \leq x \leq 10 + \frac{a}{2} : 10 - \frac{b}{2} \leq y \leq 10 + \frac{b}{2} \\ 0 & : otherwise \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the rectangular pulse. Now, the 2-dimensional cross-correlation function,  $r_{x,y}$ , applied to these images can be

calculated as follows. The Fourier transform of  $f_1(x, y)$ ,  $F_1(k_x, k_y)$  is

$$F_1(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (87)$$

$$F_1(k_x, k_y) = \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_{-\frac{a}{2}}^{\frac{a}{2}} I_0 e^{-i2\pi(k_x x + k_y y)} dx dy \quad (88)$$

$$F_1(k_x, k_y) = I_0 \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_{-\frac{a}{2}}^{\frac{a}{2}} e^{-i2\pi k_x x} e^{-i2\pi k_y y} dx dy. \quad (89)$$

To evaluate this integral, let

$$u = -i2\pi k_x x \quad (90)$$

and

$$-\frac{1}{i2\pi k_x} du = dx. \quad (91)$$

One can evaluate the x-integral by this substitution and the Euler's formula, as discussed in the previous section. Then,  $F_1(k_x, k_y)$  becomes

$$F_1(k_x, k_y) = \frac{I_0}{\pi k_x} \sin(a\pi k_x) \int_{-\frac{b}{2}}^{\frac{b}{2}} e^{-i2\pi k_y y} dy. \quad (92)$$

Similarly, let

$$u = -i2\pi k_y y \quad (93)$$

and

$$-\frac{1}{i2\pi k_y} du = dy. \quad (94)$$

One can evaluate the y-integral by this substitution and the Euler's formula, as the x-integral. Then,  $F_1(k_x, k_y)$  can be expressed as

$$F_1(k_x, k_y) = \frac{I_0}{\pi^2 k_x k_y} \sin(a\pi k_x) \sin(b\pi k_y). \quad (95)$$

Next, one can find the Fourier transform of  $f_2(x, y)$ ,  $F_2(k_x, k_y)$ , similar to the way for finding  $F_1(k_x, k_y)$ . The Fourier transform of  $f_2(x, y)$ ,  $F_2(k_x, k_y)$  is

$$F_2(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_2(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (96)$$

$$F_2(k_x, k_y) = \int_{10-\frac{b}{2}10-\frac{a}{2}}^{10+\frac{b}{2}10+\frac{a}{2}} I_0 e^{-i2\pi(k_x x + k_y y)} dx dy \quad (97)$$

$$F_2(k_x, k_y) = I_0 \int_{10-\frac{b}{2}10-\frac{a}{2}}^{10+\frac{b}{2}10+\frac{a}{2}} e^{-i2\pi k_x x} e^{-i2\pi k_y y} dx dy. \quad (98)$$

To evaluate this integral, let

$$u = -i2\pi k_x x \quad (99)$$

and

$$-\frac{1}{i2\pi k_x} du = dx. \quad (100)$$

One can evaluate the x-integral by this substitution and the Euler's formula, as discussed in the previous section. Then,  $F_2(k_x, k_y)$  becomes

$$F_2(k_x, k_y) = \frac{I_0 e^{-i20\pi k_x}}{\pi k_x} \sin(a\pi k_x) \int_{-\frac{b}{2}}^{\frac{b}{2}} e^{-i2\pi k_y y} dy. \quad (101)$$

Similarly, let

$$u = -i2\pi k_y y \quad (102)$$

and

$$-\frac{1}{i2\pi k_y} du = dy. \quad (103)$$

One can evaluate the y-integral by this substitution and the Euler's formula, as the x-integral. Then,  $F_2(k_x, k_y)$  can be expressed as

$$F_2(k_x, k_y) = \frac{I_0 e^{-i20\pi(k_x+k_y)}}{\pi^2 k_x k_y} \sin(a\pi k_x) \sin(b\pi k_y). \quad (104)$$

Now taking complex conjugate of  $F_2(k_x, k_y)$  gives

$$F_2^*(k_x, k_y) = \frac{I_0 e^{i20\pi(k_x+k_y)}}{\pi^2 k_x k_y} \sin(a\pi k_x) \sin(b\pi k_y). \quad (105)$$

The cross-correlation function  $r_{x,y}$  is defined as

$$r_{x,y} = F^{-1} \left[ \frac{F_1(k_x, k_y) F_2^*(k_x, k_y)}{S_1 S_2} \right] \quad (106)$$

or

$$r_{x,y} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{F_1(k_x, k_y) F_2^*(k_x, k_y) e^{i2\pi(xk_x+yk_y)} dk_x dk_y}{S_1 S_2}, \quad (107)$$

where  $F_1(k_x, k_y)$  is the Fourier transform of  $f_1(x, y)$ ,  $F_2^*(k_x, k_y)$  the complex conjugate of the Fourier transform of  $f_2(x, y)$ ,  $S_1$  the standard deviation of  $f_1(x, y)$ ,  $S_2$  the standard deviation of  $f_2(x, y)$ , and  $F^{-1}$  represents the inverse Fourier transform. Substituting  $F_1(k_x, k_y) = \frac{I_0}{\pi^2 k_x k_y} \sin(a\pi k_x) \sin(b\pi k_y)$  and  $F_2^*(k_x, k_y) = \frac{I_0 e^{i20\pi(k_x+k_y)}}{\pi^2 k_x k_y} \sin(a\pi k_x) \sin(b\pi k_y)$  into this equation, one has

$$r_{x,y} = \frac{1}{S_1 S_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \frac{I_0}{\pi^2 k_x k_y} \right] \sin(a\pi k_x) \sin(b\pi k_y) \left[ \frac{I_0 e^{i20\pi(k_x+k_y)}}{\pi^2 k_x k_y} \right] \sin(a\pi k_x) \sin(b\pi k_y) e^{i2\pi(xk_x+yk_y)} dk_x dk_y \quad (108)$$

which can be reduced to

$$r_{x,y} = \frac{I_0^2}{\pi^4 S_1 S_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \frac{e^{i2\pi(x+10)k_x} \sin^2(a\pi k_x)}{k_x^2} \right] \left[ \frac{e^{i2\pi(y+10)k_y} \sin^2(b\pi k_y)}{k_y^2} \right] dk_x dk_y. \quad (109)$$

To evaluate this integral, let  $\alpha = a\pi$ ,  $\beta = \pi(x+10)$ ,  $\gamma = b\pi$ , and  $\delta = \pi(y+10)$

Then, one has

$$r_{x,y} = \frac{I_0^2}{\pi^4 S_1 S_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ \frac{e^{i2\beta k_x} \sin^2(\alpha k_x)}{k_x^2} \right] \left[ \frac{e^{i2\delta k_y} \sin^2(\gamma k_y)}{k_y^2} \right] dk_x dk_y. \quad (110)$$

As described in the previous section, we use the Euler's formula, some trigonometric identities, and contour integration of  $I = \int_0^{\infty} \frac{\sin(\lambda x)}{x} dx$ , where  $\lambda$  is a positive constant. Then, one has

$$\int_{-\infty}^{\infty} \frac{e^{i2\beta k_x} \sin^2(\alpha k_x)}{k_x^2} dk_x = \pi(\alpha - \beta) \quad (111)$$

and

$$\int_{-\infty}^{\infty} \frac{e^{i2\delta k_y} \sin^2(\gamma k_y)}{k_y^2} dk_y = \pi(\gamma - \delta), \quad (112)$$

where  $\beta > 0$ ,  $\alpha - \beta > 0$ ,  $\delta > 0$  and  $\gamma - \delta > 0$ . Using these expressions, the cross-correlation function  $r_{x,y}$  becomes

$$r_{x,y} = \left[ \frac{I_0^2}{\pi^4 S_1 S_2} \right] \pi(\alpha - \beta) \pi(\gamma - \delta) \quad (113)$$

$$r_{x,y} = \left[ \frac{I_0^2}{\pi^2 S_1 S_2} \right] (\alpha - \beta)(\gamma - \delta), \quad (114)$$

where  $\alpha = a\pi$ ,  $\beta = \pi(x + 10)$ ,  $\gamma = b\pi$ , and  $\delta = \pi(y + 10)$ . Finally, defining  $x_{lag} = -x$ ,  $y_{lag} = -y$ , and imposing the conditions  $\beta > 0$ ,  $\alpha - \beta > 0$ ,  $\delta > 0$  and  $\gamma - \delta > 0$ , the cross-correlation function  $r_{x_{lag}, y_{lag}}$ , for the 2-D rectangular pulse, can be expressed as

$$r_{x_{lag}, y_{lag}} = \begin{cases} \left( \frac{I_0^2}{S_1 S_2} \right) [a + (x_{lag} - 10)][b + (y_{lag} - 10)] & : 10 - a \leq x_{lag} \leq 10 : 10 - b \leq y_{lag} \leq 10 \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a - (x_{lag} - 10)][b - (y_{lag} - 10)] & : 10 \leq x_{lag} \leq 10 + a : 10 \leq y_{lag} \leq 10 + b \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a + (x_{lag} - 10)][b - (y_{lag} - 10)] & : 10 - a \leq x_{lag} \leq 10 : 10 \leq y_{lag} \leq 10 + b \\ \left( \frac{I_0^2}{S_1 S_2} \right) [a - (x_{lag} - 10)][b + (y_{lag} - 10)] & : 10 \leq x_{lag} \leq 10 + a : 10 - b \leq y_{lag} \leq 10 \end{cases}$$

where  $I_0$  is the peak backscatter intensity of the given rectangular pulse,  $a$  is the width of the rectangular pulse,  $b$  is the height of the rectangular pulse,  $S_1$  is the standard deviation of  $f_1(x, y)$ , and  $S_2$  is the standard deviation of  $f_2(x, y)$ .