

ANALYSIS, COMPARISON AND MODIFICATION OF  
VARIOUS PARTICLE IMAGE VELOCIMETRY (PIV)  
ALGORITHMS

A Thesis

by

CARLOS EDUARDO ESTRADA PEREZ

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Nuclear Engineering

ANALYSIS, COMPARISON AND MODIFICATION OF  
VARIOUS PARTICLE IMAGE VELOCIMETRY (PIV)  
ALGORITHMS

A Thesis

by

CARLOS EDUARDO ESTRADA PEREZ

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

---

Yassin A. Hassan  
(Chair of Committee)

---

William H. Marlow  
(Member)

---

W. E. Burchill  
(Head of Department)

---

K. Annamalai  
(Member)

December 2004

Major Subject: Nuclear Engineering

# ABSTRACT

Analysis, Comparison and Modification of  
Various Particle Image Velocimetry (PIV)

Algorithms. (December 2004)

Carlos Eduardo Estrada Pérez, B.S., Universidad Autónoma Metropolitana (UAM)

Chair of Advisory Committee: Dr. Yassin A. Hassan

A program based on particle tracking velocimetry (PTV) was developed in this work. The program was successfully validated by means of artificial images where parameters such as radius, concentration, and noise were varied in order to test their influence on the results. This program uses the mask cross correlation technique for particle centroid location. The sub-pixel accuracy is achieved using two different methods, the three point Gaussian interpolation method and the center of gravity method. The second method is only used if the first method fails. The object matching algorithm between frames uses cross correlation with a non binarized image.

A performance comparison between different particle image velocimetry (PIV) and PTV algorithms was done using the international standard PIV challenge artificial images. The best performance was obtained by the program developed in this work. It showed the best accuracy, and the best spatial resolution by finding the larger number of correct vectors of all algorithm tested.

A procedure is proposed to obtain error estimates for real images based on errors calculated with experimental ones. Using this procedure a real PIV image with 20% noise has an estimated average error of 0.1 pixel.

Results of the analysis of 200 experimental images are shown for the two best PTV algorithms.

# TABLE OF CONTENTS

	Page
INTRODUCTION . . . . .	1
Literature Review . . . . .	1
Theoretical Analysis . . . . .	1
Numerical Techniques . . . . .	2
Experimental Research . . . . .	2
Objective . . . . .	7
PROCEDURE . . . . .	9
Generation of Images . . . . .	9
Home Made Artificial Images . . . . .	9
International Standard PIV Artificial Images . . . . .	13
Experimental Images . . . . .	15
Image Processing . . . . .	16
Histogram . . . . .	16
Spatial Domain Methods . . . . .	17
IMAGE ANALYSIS . . . . .	23
MaskPTV Program . . . . .	23
MaskPTV Object Identification and Centroid Location . . . . .	23
Mask Cross Correlation Algorithm . . . . .	24
MaskPTV Object Matching or Particle Tracking . . . . .	29
Fspots & Tracking Program . . . . .	33
AkaPTV Program . . . . .	34
RESULTS AND DISCUSSION . . . . .	35
Home Made Artificial Images . . . . .	35
Different Object Radius . . . . .	35
Different Concentrations . . . . .	37
Different Noise Percentage . . . . .	40
PIV Challenge Results . . . . .	43
PIVSleuth Results . . . . .	45
MaskPTV Results . . . . .	46
FspotsPTV Results . . . . .	47

	Page
AkaPTV Results . . . . .	48
Comparison Between Algorithms . . . . .	49
Experimental Results . . . . .	53
Velocity Fields . . . . .	54
Analysis of a Single Field . . . . .	57
Statistical Analysis . . . . .	59
CONCLUSIONS . . . . .	64
REFERENCES . . . . .	66
APPENDIX A . . . . .	68
VITA . . . . .	69

# LIST OF TABLES

	Page
1    MaskPTV object classification parameters . . . . .	55
2    MaskPTV object tracking parameters . . . . .	55
3    AkaPTV object classification parameters . . . . .	55
4    AkaPTV object tracking parameters . . . . .	56

# LIST OF FIGURES

FIGURE	Page
1 (a) Artificial image with 2000 particles and 10% Gaussian noise, (b) artificial image with 2000 particles, 200 bubbles and 10% Gaussian noise, (c) experimental single phase image, (d) experimental two phase image . . . . .	11
2 Examples of home made artificial images directional fields (a) strong vortex with high velocity gradient in the center, (b) explosion-like field (c) simulation of two phases, each phase following different directional fields . . . . .	13
3 Okamoto standard images (a) $t = 0$ , (b) $t = \Delta t$ , (c) velocity field . . . . .	14
4 PIV challenge images (a) image B001_1, small particles high concentration, (b) image B006_1 large particles low concentration, (c) approximated velocity field . . .	15
5 A 3 x 3 neighborhood about a point $(x, y)$ in an image . . . . .	18
6 Illustration of the histogram-equalization method. (a) original histogram. (b) transformation function. (c) equalized histogram . . . . .	20
7 Illustration of the histogram-equalization method. (a) original image. (b) equalized image . . . . .	21
8 Shape modifier parameters influence on the object mask (a) $A=1$ , $B=1$ . (b) $A=0.5$ , $B=0.5$ . (c) $A=2.0$ , $B=2.0$ . (d) $A=1$ , $B=0.5$ . (e) $A=0.5$ , $B=1$ . . . . .	25
9 Cross correlation coefficient plane . . . . .	26
10 Thresholding influence in object area detection. Colors white = pixels with a gray scale value larger than GST and $C_1$ lower than 0.0, blue = pixels with $C_1$ value from 0.0 to 0.2, green = pixels with $C_1$ value form 0.2 to 0.4, yellow = pixels with $C_1$ value from 0.4 to 0.6 and red = pixels with $C_1$ value from 0.6 to 1.0 . . . . .	27
11 Top-left assignment algorithm, the numbers represent the object identification . . .	28
12 Search window . . . . .	30
13 Example of tracking coefficient windows (TCW) and object matching procedure . .	33
14 Different particle radius images and related RMS error . . . . .	36
15 Yield, reliability, and RMS as a function of object radius . . . . .	37
16 Different concentration images and the resulting centroid error . . . . .	38
17 Yield, reliability and RMS of object centroid estimation as a function of object concentration . . . . .	39
18 Population of objects and RMS depending on the $C_1$ . . . . .	40

FIGURE	Page
19 RMS depending on the $C_1$ and noise percentage . . . . .	41
20 Population depending on $C_1$ and noise percentage . . . . .	41
21 RMS, yield, and reliability of vectors location estimation for different noise percentage	42
22 Number of vectors found by different image analysis techniques using PIV challenge images . . . . .	43
23 RMS of vectors located by different image analysis techniques using PIV challenge images . . . . .	44
24 RMS probability density function of the displacement for different image concentra- tion and sizes. Analysis with PIVSleuth . . . . .	45
25 RMS probability density function of the displacement for different image concentra- tion and sizes. Analysis with MaskPTV . . . . .	47
26 RMS probability density function of the displacement for different image concentra- tion and sizes. Analysis with FspotsPTV . . . . .	48
27 RMS probability density function of the displacement for different image concentra- tion and sizes. Analysis with AkaPTV . . . . .	49
28 RMS probability density function comparison between the different image analysis techniques for case (a) B001, (b) B003, (c) B005 . . . . .	50
29 RMS probability density function comparison between the different image analysis techniques for case (a) B002, (b) B004, (c) B006 . . . . .	51
30 Velocity field obtained with (a) PIVSleuth, (b) MaskPTV for B001 case and (c) PIVSleuth, (d) MaskPTV for B006 case . . . . .	52
31 Velocity field obtained with (a) AkaPTV and (b) FspotsPTV for B001 case and (c) AkaPTV and (d) FspotsPTV for B006 case . . . . .	53
32 Velocity field of the two phases and bubble location using (a) MaskPTV algorithm, and (b) AkaPTV algorithm . . . . .	56
33 Comparison of the resulting velocity fields between (b) AkaPTV program and (c) MaskPTV program obtained with the analysis of (a) original image . . . . .	57
34 Instantaneous $u$ velocity distribution vs $y$ coordinate . . . . .	57
35 Filtered $u$ instantaneous velocity distribution vs $y$ coordinate . . . . .	58
36 $C_2$ distribution over the $y$ axis . . . . .	59
37 (a) Influence of the $C_2$ over the average in time of $u$ velocity profile, (b) influence of different $C_2$ over the number of vectors found on the whole set . . . . .	60



FIGURE	Page
38 (a) Velocity profile $u$ obtained from MaskPTV and AkaPTV programs, (b) velocity profile $v$ obtained from MaskPTV and AkaPTV programs . . . . .	61
39 (a) $u'$ turbulence intensity from MaskPTV and AkaPTV programs, (b) $v'$ turbulence intensity obtained from MaskPTV and AkaPTV programs . . . . .	62
40 $u'v'$ Reynolds stresses obtained from MaskPTV and AkaPTV programs . . . . .	63

# INTRODUCTION

## LITERATURE REVIEW

Turbulence has been investigated for more than a century, and still there is not a complete understanding of the phenomena governing it. Approaches adopted to study turbulence are directed by theoretical, numerical, and experimental techniques, and considerable progress has been made by the contribution of these techniques.

### Theoretical Analysis

In the development of a theory for turbulence, there are three major approaches which have received considerable attention, namely the statistical, phenomenological, and numerical approaches.

In the statistical theory of turbulence all quantities are decomposed into the sum of a time averaged part and a fluctuating part and then substituted into the Navier-Stokes equations and the mass and energy conservation laws. The equations are then averaged in order to obtain a set of equations that describes the mean behavior of turbulent flows (Tennekes and Lumley 1972). Unfortunately, the nonlinearity of the equations results in more unknowns than governing equations. Further relationships can be developed using various moments of the conservation laws, but these additional manipulations inevitably introduce still more unknowns. This is the “Closure Problem” of the Statistical Theory of Turbulence. To attack this problem several approaches have been suggested: a mathematical approach, where arbitrary simplifications of the existing equations are made by dropping terms or by expressing them in convenient forms suggested by analogy with laminar flow; a physical approach where specific models of turbulent activity were used to establish new relationships; and a combined approach where both mathematical and physical arguments are used (Frost and Mouldin 1977) .

Phenomenological or semi-empirical theories combine heuristic arguments and empirical data to predict the gross phenomena arising from turbulent activity. The goal of the phenomenological theories is to obtain a functional dependency for the Reynolds stress directly. Such theories are: Boussinesq’s Theory, Prandtl’s Mixing Length Theory, Taylor’s Vorticity-Transport Theory, Von Karman’s Similarity Hypothesis, etc.

The phenomenological theories assume some simple mechanism for turbulence and then derive from this the desired relation for the Reynolds stress. Unfortunately, none of them gives a complete physical picture, nor a comprehensive mathematical model (Brodkey 1967).

### **Numerical Techniques**

Conceptually, numerical techniques can be used to analyze almost any flow field. In a turbulent flow field, where both small and large scales of motion coexist, a mesh size small enough to resolve the smaller scales is needed. At the same time, the number of grid points has to be large enough in order to include the large scales of motion as well. Considering that the difference in magnitude between the small and large scales for high Reynolds numbers can be of the order of  $10^4$  to  $10^5$ , the resulting number of grid points needed to solve the problem is extremely large and makes this approach unfeasible except for very low Reynolds numbers. An example of such flow field is the turbulent channel flow studied by Mossi and Sagaut (2003). Their numerical simulation required 317,135 grid points (41 x 65 x 119 in x, y, z) for a Reynolds number of 3000. Even the best and fastest computers today are not able to handle similar Navier-Stokes direct simulations with the number of grid points required for engineering Reynolds numbers.

Although the recent theoretical and computational developments of turbulence have provided many insights into its mechanisms, the majority the researchers today are still using experimental information to validate existing theories and simulations.

### **Experimental Research**

The shortcomings of all the methods and techniques described above has put a great deal of emphasis on experimental research. Researchers have been using both qualitative methods such as flow visualization, and quantitative methods such as hot wire anemometry, laser Doppler velocimetry, streak photography, pulsed laser velocimetry, and various other imaging techniques, to try to understand the mechanisms of turbulence. The next sections give a short introduction to each of these fluid analysis techniques.

**Flow Visualization Techniques** Flow visualization techniques provide valuable information about the behavior of the entire flow field but are not usually intended to provide quantitative velocity data. Using flow visualization, the fluid in a large region can be simultaneously observed. This allows the researcher to get a vivid, global and detailed picture of the physical process taking place in the flow, but the interpretation of such data is very subjective. Nevertheless, flow visualization has been a useful tool in the hands of turbulent flow researchers for many years and

has brought many breakthrough discoveries. Examples of such discoveries include: the ejections and bursts that occur locally and randomly in space and time in a turbulent wall flows, (Corino and Brodkey 1969), the hairpin vortices in the boundary layer (Head and Bandyopadhyay 1981) and the coherent structures which dominate the mixing layer (Brown and Roshko 1974). Other excellent studies were done by Offen and Kline (1974) where they used two basic flow visualization techniques, dye injection and hydrogen bubble generation to study the turbulent boundary layer over a flat plate, and by Nychas et al (1973) where they used high speed photography to record the motion of very small solid particles in the outer region of a turbulent boundary layer along a flat plate. Praturi and Brodkey 1978 used stereoscopic photography and they identified the axial vortex motions in the turbulent wall region because they were able to catch the three-dimensional aspects of the flow. Up until recently, very little quantitative information could be extracted from the flow visualization images because of the time required to manually extract the data.

**Hot Wire Anemometer** For more quantitative measurements, hot wires have been used in many different configurations by researchers for a long time. The hot wire anemometer is basically a thermal transducer. It consists of a very thin, short, metal wire, heated by electrical current. Its operation relies on the fact that the flowing fluid cools the wire, causing a temperature drop and that temperature drop is related to the instantaneous velocity of the flow field at the point of the measurement (Hinze 1975). The hot wire system has two modes of operation. The first is the constant current mode, where the current in the wire is kept constant and variations in the wire resistance caused by the flow are measured by monitoring the voltage drop variations across the hot wire, the second is the constant temperature mode, where the metal wire is placed in a feedback circuit which tends to maintain the wire at constant resistance and hence at constant temperature (Perry 1982). While being widely used, hot wire anemometry suffers from many drawbacks. The small size of the wire (of the order of  $5\ \mu m$  in diameter) makes it unsuitable for use in harsh environments such as combustion flows (Goldstein 1983) and it can not be used to study practical problems such as separated flows with high turbulence intensities and flow reversals. Hot wire anemometers are also sometimes inaccurate to obtain measurements very close to a solid boundary. Multiple hot wires can be used simultaneously to determine multiple velocity components but such probes produce more disturbances in the flow field and the measurement volume becomes large with respect to the smallest turbulence scales. Furthermore, measurements are made at a single point in space and spatial information would be useful to further understanding of turbulence. The pulsed wire anemometry technique was developed by Bradshaw (1971). In essence, the probe consists of three thin wires mounted short distances apart. The two “receiver” wires are mounted on either

side of the “transmitting” wire at right angles to it. The “transmitting” wire is heated periodically with a pulsed current and the resulting streaks of heated fluid are convected downstream by the flow. The “receiving” wires operate in the same manner as a normal hot wire anemometer. The time elapsed between the release of a hot streak by the “transmitting” wire and its detection by either of the “receiver” wires is recorded electronically and then converted into velocity. The sign of the velocity is determined by noting which of the two “receiver” wires receives the pulse. This technique is independent of the properties of the fluid and gives direct measurement of the velocity. It can also be used to measure velocities in very high turbulence regions. However, it suffers from large probe interference, poor spatial resolution and a very poor frequency response because the time interval between pulses can not be less than the smallest passage time between wires.

**Laser Doppler Velocimetry** The first laser Doppler velocimetry (LDV) was described by Yeh and Cummins (1964) and since then, there have been numerous developments of both the optics and the processing electronics of the system. The LDV technique utilizes the Doppler effect to measure instantaneous velocity of small particles embedded in the flow. When a particle passes through the measurement volume formed by focusing two laser beams, it scatters the light. The scattering light is at a different frequency from that of the incident beam and the particle velocity can then be determined by measuring the Doppler shift of the scattered light. LDV presents some advantages over the hot wire anemometry techniques such as, no flow interference (since the probing is purely optical) and direct velocity measurement without calibration regardless of the velocity field.

Furthermore, the technique is quite accurate, but its spatial resolution is not very good and its costs are high. The technique relies on seeding the flow with small scattering particles.

To overcome the limitations of single point measurements, new techniques have been devised which draw from the advantages of flow visualization while providing quantitative information about the spatial structure of the flow. The development of relatively inexpensive digitizing cameras and digital computers, as well as the development of elaborate software, has now allowed the researchers to develop techniques to extract full field, quantitative information from flow visualization of images.

**Streak Photography** Streak photography is the oldest and simplest method of experimental multi-point velocity measurements. The flow is seeded with particles, and the fluid velocity can be measured as a function of time by tracking the particles down through a succession of multiple exposure recordings or multiple frame techniques. Dimotakis et al (1981) used sawdust as tracer particles, and using streak photography, they investigated a turbulent shear layer by measuring two components of the velocity flow field. Sheet lighting is necessary to define the plane of fluid motion.

This can present a problem due to the fact that when the particles move out of the light sheet in a three-dimensional field, their streaks are wrongly interpreted as low velocities. A solution to this problem was given by Kobayashi and Ishihara (1985). They used fluorescent particles stimulated by a light pulse. Because of the fluorescence, the particles did not have to remain within the light sheet, and their fluorescent tails could be observed for the complete duration of the exposure, enabling the determination of their velocities and directions (Adrian 1986). Velocity measurements using streak photography have a large associated error when the length of the streak is small. Also, when the mean distance between tracers is of the same order of magnitude as the distance a tracer particle travels during the exposure time, streak photography fails to provide accurate results. Thus, the tracer concentration is usually kept small resulting in velocity measurements with poor spatial density. Involved interpolating methods have been designed in order to increase the spatial resolution of the technique, but their validity is questionable if the distance between the existing data points is larger than the turbulent flow scales (Lourenco 1986). Multiple frame techniques such as conventional motion pictures or multiple exposures on the same frame can be used. Nishino and Kasagi (1985) filmed the motion of small patches of hydrogen bubbles in a turbulent boundary layer.

**Pulsed Laser Velocimetry** The availability of high power laser sources together with fast digital processors led to the development of sophisticated whole field velocimetry techniques such as laser speckle velocimetry (LSV), particle image velocimetry (PIV) and particle tracking velocimetry (PTV). These techniques provide the simultaneous visualization of the two-dimensional streamline pattern in unsteady flows as well as the quantification of the velocity field over the entire plane. LSV and PIV have been established as two operating modes of the same method (Adrian 1984) and they differ simply by the concentration of tracers used to seed the fluid.

The common measuring principle behind these methods is that instantaneous fluid velocities can be evaluated by recording the position of images produced by small tracers, suspended in the fluid, at successive time instants. The underlying assumption is that these tracers closely follow, with minimal lag, the fluid motion. This assumption holds true for a wide variety of flows of interest provided that the tracers are small enough and/or that their density approaches that of the fluid.

Besides their common goal LSV and PIV, on one hand, and PTV on the other hand do not share the same historical development and practice. An important difference is that in LSV and PIV, the concentration of tracers is rather high and the measurement of the “local” fluid velocity results from an average over many tracers contained in a measurement volume (cell). The cell is usually regularly spaced and its size determines the spatial resolution. This is in contrast with

PTV where the velocity is determined at random locations using the images produced by a single tracer. PTV has evolved as a method of extracting quantitative information from conventional flow visualization data, such as streak photography or multiple exposure photography Kobayashi and Ishihara (1985). While this methods excel as a means for fast and easy mapping of the flow basic structures, they fail in providing an accurate velocity field map with high spatial density.

Typically, the evaluation of the particulate displacements directly from their images, such as evaluating streak length or spacing between successive images of the same tracer, requires processing large amounts of data and rather sophisticated software. In addition, the measurement errors become large when the mean distance between tracers is of the same order of magnitude of the streak length, or the spacing between successive particle images. As a consequence, the tracer density is kept small, resulting in velocity measurements with poor spatial resolution. The problem of having sparse velocity information at random locations has been addressed with the introduction of rather sophisticated interpolating schemes. However, the validity of this approach is questionable considering that in most cases the spacing between data points is larger than the flow scales, and the velocity may be interpolated at best with first order accuracy (Riethmuller 2001).

The overall accuracy of PLV techniques depends on the accuracies which can be achieved in the photographic procedure and the Imaging techniques to implement. Work to study and improve the photographic procedure was done by Russ et al (1988) where parameters like illumination and contrast were changed to obtain better images, the minimization of the obstruction of the grid used and the visibility of the particles was investigated. Also, the color of the particles was analyzed by digitizing the images and obtaining the average R, G and B values out of them.

Imaging techniques are used after an image has been obtained using one of the methods described above. They are basically divided into two categories: image processing, which is concerned with the generation of new images from already existing ones (filtering, noise reduction, edge enhancement, etc.) and image analysis, which is the task of obtaining useful information from the image. Image analysis is subject to the individual interests of each researcher an it includes image enhancement, particle detection, particle tracking, image cross correlation, velocity averaging and evaluation of error.

Nowadays, PIV and PTV techniques have gain big popularity between turbulence researches around the world because of their improved overall performance, several projects of cooperative nature had been created to set standards to evaluate the accuracy and performance of different image analysis algorithms. One example of the use of the standardized evaluation is the work of Tetsuo et al (2001). They used an iterative cross correlation PTV algorithm, and report an average

error in the particle centroid position detection of 0.05 pixel and an average error in the velocity vector of 0.05 pixel, using artificial standard images (Okamoto et al 1998).

Works to reduce the disadvantages inherent of PIV and PTV have been developed; hybrid algorithms combine PIV and PTV characteristics to obtain the so called hi resolution particle velocimetry (HIRES-PV) (Van der Plas and G Bastians 1998). HIRES-PV uses PIV to give a preliminary estimate of the velocity that will be used by the PTV tracking routine to search smaller regions for a particle match. Thus seeding density can be increased and the spatial resolution, compared with a traditional PTV algorithm, is improved, the added noise due to interpolation techniques is reduced, and the final result is better than typical PIV techniques because PTV is inherently less affected by the presence of large displacement gradients.

## OBJECTIVE

Several efforts have been directed to determine the error induced by different imaging techniques with the use of artificial images where all objects and fluid characteristics are known and controlled, and the error associated with calculated values such as particle size, particle centroid location and particle velocity is easily obtained. When dealing with real experimental images this error analysis must be indirectly calculated using, as a base, a previous error analysis of artificial images that mimic the experimental ones. Different artificial images are needed for each different experiment.

Imaging technique parameters will depend on the researcher goals and interests. For example, when working with turbulent two-phase flow, the researcher might be interested only in the information of the liquid phase; therefore, the parameters used to enhance the image and to obtain an object location will differ from the parameters used when trying to find information of the gaseous phase. Therefore, different algorithms of imaging techniques must be tested to obtain their optimal parameters in order to extract real and accurate quantitative information from the images.

This work is intended to answer some of the previous questions. Several imaging techniques will be tested, one to obtain an accurate error value of the resulting velocity field, secondly to choose which imaging technique is more suitable for a given experiment and thirdly, to modify and improve the currently used imaging techniques to satisfy a specific study of a turbulent two-phase flow. Hence, the objective of this research is summarized as:

- Use the international standard artificial images and the home made artificial images to test and compare different imaging techniques.
- Analyze the influence of the imaging techniques parameters on the results.



- Analyze the influence of objects characteristics like size, concentration and noise on the results.
- Develop a stand alone PTV windows application capable to obtain velocity vectors with high accuracy.
- Suggest a systematical method in order to obtain the best statistical results from experimental images.
- To obtain estimated experimental errors from the analysis of artificial images.

## PROCEDURE

This section is a step by step description of the procedure used in order to meet the objectives of the study.

### GENERATION OF IMAGES

Three types of images were used in this research: home made artificial images, standard artificial images and experimental images. The details of the generation or acquisition of them is given in the next sections.

#### Home Made Artificial Images

Home made artificial images were generated in order to mimic a broad class of real images obtained from the actual implementation of PIV by simulating the behavior of specific flow fields to test the performance of the different imaging techniques under extreme cases likely to happen in real experiments. Every image is computer digitized with a sampling of 500 x 500 pixels and 8 bit depth, where the image sampling is the display area in pixels, and the gray-scale represent the maximum number of allowed gray levels that a pixel can have; the resolution (i.e. the degree of discernible detail) of an image is strongly dependent on image sampling and gray scale. The more these parameters are increased, the closer the digitalized array will approximate the original image. However, the resolution is limited by several factors. The better the resolution is, the greater the storage space and processing requirements are. Furthermore, the resolution selected to produce these images is such that they will mimic the output pictures from the CCD camera used in this work, having both approximately the same average dimensions and behavior of the objects represented in the pictures. In this array of 500 x 500 pixels, the entities represented in the image are hypothetical particles and bubbles embedded within an ideal black background.

The specialized option of having different objects within an image gives the opportunity of creating “bubbles” that will differ in size, gray scale distribution, shape, concentration, and velocity from the objects representing particles. This is the only feature that the international standard images does not have, and therefore this is one the main reasons why the home made images generation is justified, the final goal is to evaluate the performance of imaging techniques when analyzing experimental images of a two phase turbulent flow.

**Object Generation** As mentioned before, the objects generated within the artificial images are essentially three: objects that represent solid particles, bubbles and background.

The background is considered to be fluid, having the same size as the image, 500 x 500 pixels. Ideally, the gray level for each pixel representing the background can be set to zero (black), but in a real PIV picture the background will be affected by multiple factors: experimental set up, photographic processes, reflections from the fluid container, digitizers and even the environment where the picture is taken. All these factors are “noise” that reduce the image quality. In order to imitate a realistic background artificial images are modified by adding Gaussian additive noise.

The gray scale intensity of pixels representing an object (particle or bubble) will follow a “Gaussian-like” function expressed by:

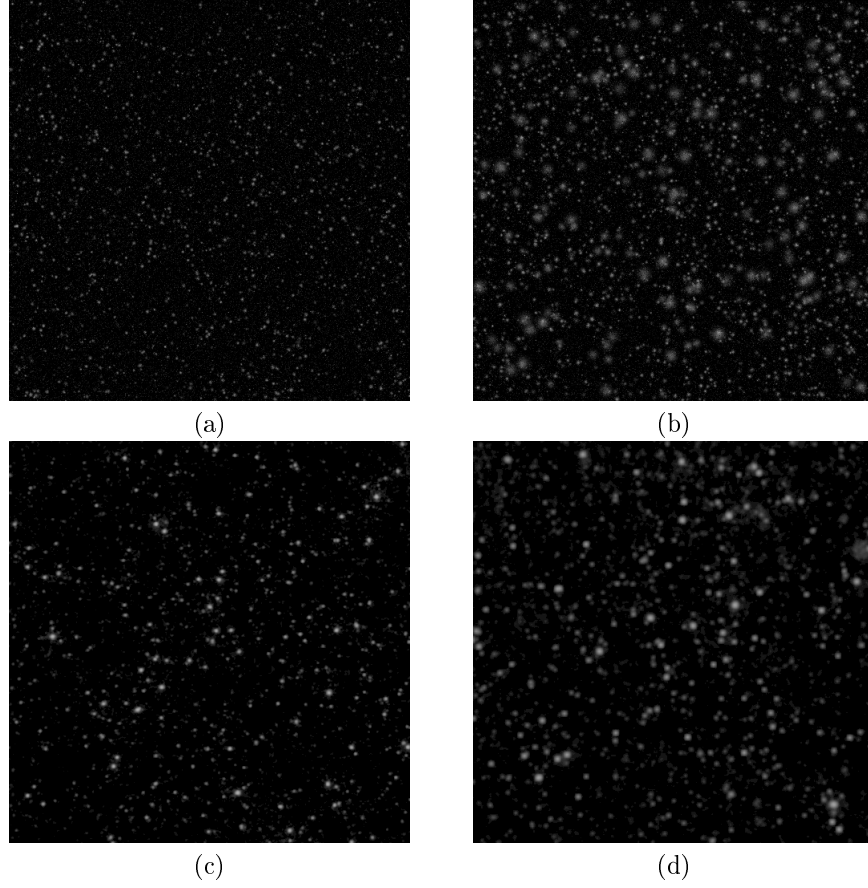
$$I(x, y) = I_0 \exp \left( -\frac{(x - x_o)^2/a^2 + (y - y_o)^2/b^2}{2d_o^2} \right) \quad (1)$$

where  $I(x, y)$  is the gray scale intensity in the position  $(x, y)$  in pixels,  $(x_o, y_o)$  is the object centroid location,  $I_0$  is the maximum intensity of the object,  $a$  and  $b$  are shape modifier parameters and  $d_o$  is the particle diameter in pixels. When a pixel is part of two objects, the value of its gray scale intensity  $I(x, y)$  is calculated as the simple average of the intensities in that pixel for each object.

The values for the previous parameters are chosen considering that artificial objects must be like real objects represented in the experimental images. Therefore, an a priori study of the experimental images must be done to obtain values for size, intensity and shape of objects. Even though real objects may have the same size and shape, they can appear differently from each other depending on their orientation on the plane where the picture is taken and the possible existence of shadows due to other particles.

Based on this broad set of assumptions, every image is generated according to a specified set of parameters. These include the number of particle ( $N_p$ ), number of bubbles ( $N_b$ ), minimum and maximum radius for particles ( $d_{r_{min}}, d_{r_{max}}$ ), minimum and maximum radius for bubbles ( $d_{r_{min}}, d_{r_{max}}$ ), minimum and maximum gray scale intensity for particles ( $I_{p_{min}}, I_{p_{max}}$ ), minimum and maximum gray scale intensity for bubbles ( $I_{b_{min}}, I_{b_{max}}$ ) and the image size in pixels on  $x$  and  $y$  direction ( $L_x, L_y$ ).

Shown in figure 1 are the similitudes of artificial and experimental images. A Gaussian noise of 10% was added to artificial images in figure 1(a) and in figure 1(b) to increase the realism. In figure 1(c) and figure 1(d) are the results of applying image processing to diminish reflections and noise inherent to the experimental processes.



**Figure 1:** (a) Artificial image with 2000 particles and 10% Gaussian noise, (b) artificial image with 2000 particles, 200 bubbles and 10% Gaussian noise, (c) experimental single phase image, (d) experimental two phase image

Since physically all values of the allowable  $x$  coordinates ( $x_o$ ) and  $y$  coordinates ( $y_o$ ) are “equally likely” to occur, the object center coordinates were generated by a random number generator. Specifically, the coordinates of the centers of each object are calculated as follows:

$$x_o(i) = \xi L_x \text{ and } y_o(i) = \xi L_y \quad (2)$$

where  $\xi$  is a uniform randomly distributed deviate which takes values between 0.0 and 1.0 ,  $x_o$  and  $y_o$  are the coordinates of the center of the object ( $i$ ). Similarly the size of the objects  $d_o$  and their maximum gray scale intensity  $I_o$  are chosen from the random number generator.

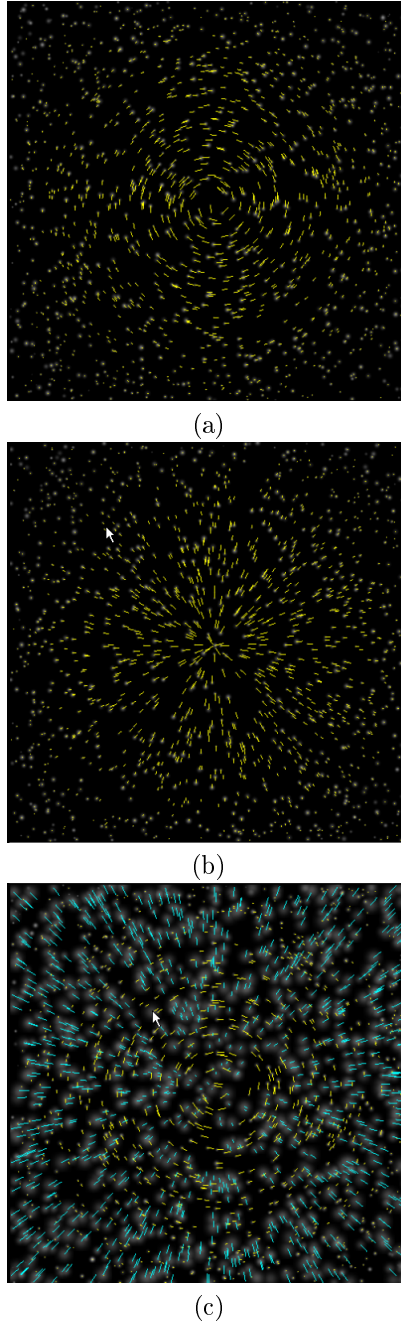
**Direction Field Generation** One of the goals of the home made artificial images is to generate velocity fields that represent challenges for the imaging techniques. The methodology employed to achieve this task is explained next.

For the direction field generation, a graphical method is used for displaying the general shape and behavior of solutions to  $y' = f(x, y)$ . This method uses Euler's lineal element algorithm, where each lineal element is a short line segment centered at the initial object center  $(x_i, y_i)$ . The ending point  $(x_f, y_f)$  of the segment is then computed by the Euler approximation:

$$x_f = x_i + t \quad y_f = y_i + tf(x_i, y_i) \quad (3)$$

A detailed example of this procedure is shown in Appendix A.

In figure 2, examples of different directional fields are shown. Figure 2(a) is the simulation of a single phase flow with 1500 particles with a radius from 0.5 to 3 pixels following a directional field with  $y' = -x/y$ . Figure 2(b) is also the simulation of a single phase flow with 1500 particles with a radius from 0.5 to 3 pixels following a directional field of with  $y' = y/x$ , and figure 2(c) is the emulation of a two phase flow, with 1000 particles with a radius from 0.5 to 1 pixels and 1000 bubbles with a radius from 1 to 3 pixels, the particles follows a directional field with  $y' = -x/y$ , and the bubbles follows a directional field with  $y' = -x/y$ . Three main characteristics make the previous images a challenge to imaging techniques. First, the fact that there exists a huge difference of velocities ranging from nearly 0.1 pixels per unit time to almost 18 pixels per unit time, makes the analysis especially complicated. Previous knowledge of the behaviour of the flow can't be used as a basis to help decrease the searching area to match particles. Secondly, the existence of counter-current is another factor that increases the probability of failure of the tracking algorithms. Third, the different characteristics between bubble and particle objects like directional field, size, and shape tests the accuracy of object identification.



**Figure 2:** Examples of home made artificial images directional fields (a) strong vortex with high velocity gradient in the center, (b) explosion-like field (c) simulation of two phases, each phase following different directional fields

### International Standard PIV Artificial Images

The objective of using international standard PIV artificial images is having a mean of comparison between the image analysis techniques studied in this work. One important characteristic of the standard artificial images is the world wide recognition and distribution. They are available

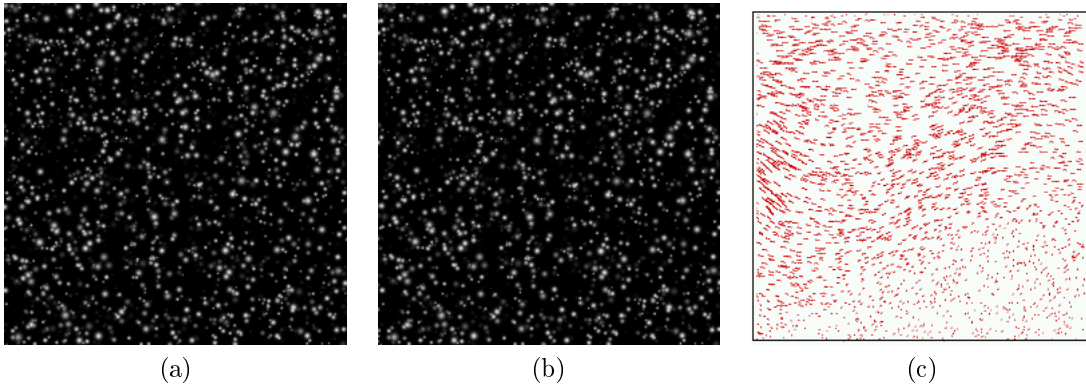
from the Internet, and the personalization of parameters for the generation of such standard images makes them even more attractive. Two main characteristics are considered in the generation of these images (Okamoto et al 2000):

- A two-dimensional laser-light sheet is illuminated in the transient/steady three-dimensional flow field, that is the two-dimensional measurement of the three-dimensional velocity field.
- The tracer particles is a micro-sphere with hypothetical scattering. The size of the particle is large enough for one to identify the individual particles.

These assumptions mostly cover any of the two-dimensional PIV techniques.

**Velocity Field** The target flow field should be precisely known for the standard images. The flow field should also include the three-dimensional velocity information. The velocity distribution calculated by using the three-dimensional large eddy-simulation (LES) code has been selected as the target flow field (Tsubokura et al 1997). The two-dimensional planar jet impinges on the wall at jet Reynolds number 6000, which corresponds to turbulence. Lots of vortexes with various scales exist in the flow field. The simulation volume is  $53B \times 10B \times 3.9B$  for the  $x$ ,  $y$  and  $z$  directions, where  $B$  is the nozzle width. The volume is divided into  $300 \times 100 \times 34$  meshes with variable mesh sizes. The flow field has been solved by the LES technique. The instantaneous three-dimensional velocity distribution at any point can be extracted from the simulation results.

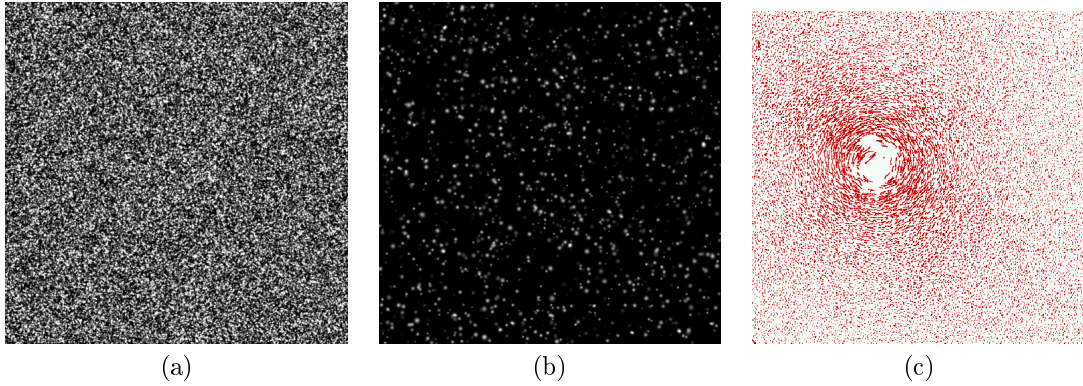
The standard images were downloaded from the web site <http://www.vsj.or.jp/piv>, and is a contribution of the Visualization Society of Japan (VSJ). Examples of these images are shown in figure 3 where a consecutive pair is shown in (a) and (b), and figure 3(c) is the velocity field.



**Figure 3:** Okamoto standard images (a)  $t = 0$ , (b)  $t = \Delta t$ , (c) velocity field

Other standard artificial images used in this work are the ones generated for the first interna-

tional PIV challenge (Okamoto *et al* 2003). The flow field is a strong vortex generated numerically, using the Japanese standard image project synthetic image generator ([www.vsj.or.jp/piv](http://www.vsj.or.jp/piv)). The particle distribution in the flow was assumed to be Gaussian. The occlusion of particle images was taken into account (the intensity of the particle images was linearly added). The camera fill factor was taken as 0.7. The particle diameters were randomly distributed following a Gaussian law. The laser light sheet intensity distribution was supposed to be Gaussian. The maximum out-of-plane velocity component was set so that the maximum displacement normal to the plane is 30% of the light sheet thickness. The images generated were  $512 \times 512$  in size with a depth of eight bits. Six different image sets were used to vary the particle image characteristics, both for PIV (small particle images: B001, B003, B005) and for PTV (large particle images: B002, B004, B006). The second varying parameter for each set was the particle concentration, figure 4 shows the first image for two extreme cases: B001 (high concentration, small particles) and B006 (large particles, low concentration). The second varying parameter for each set was the particle concentration, figure 4 shows the first image for two extreme cases: B001 (high concentration, small particles) and B006 (large particles, low concentration).



**Figure 4:** PIV challenge images (a) image B001\_1, small particles high concentration, (b) image B006\_1 large particles low concentration, (c) approximated velocity field

### Experimental Images

The experimental images used in this work come from an experiment to study drag reduction in a two-phase flow turbulent channel. The channel is made of  $12.7\text{ mm}$  thick plexiglas. Its dimensions are: length  $4.83\text{ m}$ ,  $20.5\text{ cm}$  wide and  $5.6\text{ cm}$  height. Nozzles are placed at both ends to allow quick production of a two-dimensional flow. The inlet nozzle has plastic screens and flow straighteners (plastic straws). 175 flow straighteners are set in this nozzle. Their diameter is  $4.5\text{ mm}$  internal diameter. The mesh size for the screens is about  $1.5\text{ mm}^2$ . The maximum flow currently achievable is  $2.6 \times 10^{-3}\text{ m}^3/\text{s}$ . The maximum averaged cross section velocity in the channel is  $22.7 \times 10^{-2}\text{ m/s}$ . The test section is located over 68 channel heights in the downstream direction ensuring fully



developed flow.

The illumination is given by a  $350\text{ mJ/pulse}$  Spectra Physics Nd:Yag twin-laser with a wavelength of  $532\text{ nm}$  (green range of visible light). The laser generates two  $7\text{ ns}$  pulses with a variable time span between them. The time between pulses was kept constant during all the experiments at  $1\text{ msec}$ . The laser sheet has a thickness of  $1\text{ mm}$ . A high-resolution camera was focused onto the light from the side, with the focusing lens of the camera at a distance of  $33.5\text{ cm}$  from the light sheet. For this experiments the particles used as seedings have an average diameter of  $6\text{ }\mu\text{m}$  with a density similar to the density of the water, specific gravity of 1.05. Its refraction index fulfills the PIV requirements. The particle model is Expancel 091 DU (dry-unexpanded). An example of the experimental images is shown in figure 1.

## IMAGE PROCESSING

The image processing, or image enhancement, principal objective is to process a given image so that the result is more suitable for a specific application. The word “specific” is important because it establishes at the outset that the techniques discussed are very much problem-oriented. Thus, for example, a method that is quite useful for enhancing x-ray images may not necessarily be the best approach for enhancing PIV pictures.

In this work, the image processing techniques used are equalizing, logarithmic gamma correction, averaging, smoothing and thresholding. Details of each technique and some basic concepts are explained next.

### Histogram

The histogram of an image indicates the quantitative distribution of pixels per gray-level value. It provides a general description of the appearance of an image and helps identify various components such as background, objects, and noise. The histogram is a function  $H(k)$  defined on the gray-scale range  $[0, \dots, k, \dots, 255]$  such that the number of pixels having equal gray-level value  $k$  is

$$H(k) = n_k \quad (4)$$

where  $k$  is the gray-level value,  $n_k$  is the number of pixels in an image with a gray-level value equal to  $k$ .

## Spatial Domain Methods

On the attempt of improving the original image, the histogram has to be modified with specific transformation functions. The characteristics of this function depends on the type of approach selected. There are two approaches in image processing: frequency-domain methods and spatial-domain methods. Processing techniques in the first category are based on modifying the Fourier transform of an image and are not used in this study. The spatial domain, on the other hand, refers to the image plane itself, and approaches in this category are based on direct manipulation of the pixels in an image.

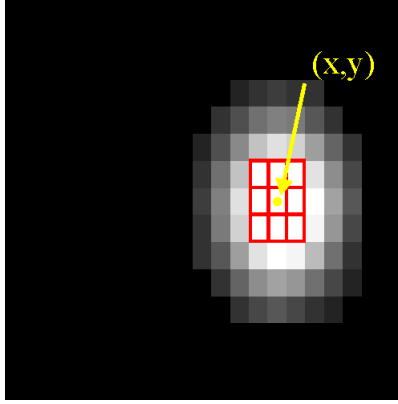
The term spatial domain refers to the aggregate of pixels composing an image, and spatial-domain methods are procedures that operate directly on these pixels. Image-processing functions in the spatial domain may be expressed as

$$g(x, y) = T[f(x, y)] \quad (5)$$

where  $f(x, y)$  is the input image,  $g(x, y)$  is the processed image, and  $T$  is an operator on  $f$ , defined over some neighborhood of  $(x, y)$ . It is also possible to let  $T$  operate on a set of input images, such as performing the pixel-by-pixel sum of  $k$  images for noise reduction.

The principal approach used in defining a neighborhood about  $(x, y)$  is to use a square or rectangular sub-image area centered at  $(x, y)$ , as shown in figure 5, the center of the sub-image is moved from pixel to pixel starting, say at the top left corner, and applying the operator at each location  $(x, y)$  to yield the value of  $g$  at that location. Although other neighborhood shapes, such as circle, are sometimes used, square arrays are by far the most predominant because of their ease of implementation.

The neighborhood about  $(x, y)$  is sometimes called kernel. The definition of the kernel will depend on the operator  $T$ , the kernel can vary in size and shape.



**Figure 5:** A 3 x 3 neighborhood about a point  $(x, y)$  in an image

The simplest form of  $T$  is when the neighborhood is  $1 \times 1$ . In this case,  $g$  depends only on the value of  $f$  at  $(x, y)$  and  $T$  becomes a gray-level transformation (also called mapping) function of the form

$$s = T[r] \quad (6)$$

where, for simplicity in notation, we use  $r$  and  $s$  as variables that denote the gray level of  $f(x, y)$  and  $g(x, y)$  at any point  $(x, y)$ .

**Averaging** One example of a gray-level transformation that is very common between PIV researches is the simple average. It is used to subtract the background noise from a set of images. In this procedure, a model of the background that has to be removed is obtained from the average of the acquired sequence of frames. The net effect is that rapidly moving particles contribute little to the pixels along their path on average, and therefore, do not appear on the background model. Only stationary noise is kept into account because it is present in every frame. The background model is then subtracted to every frame before any further processing. Sometimes the illumination or other causes of this kind of noise are slowly changing with time. In this case a moving average is performed in order to obtain a time dependent background model to be subtracted. Consider a noisy image  $g_i(x, y)$  that is formed by the addition of noise  $\eta(x, y)$  to an original image  $f_i(x, y)$ ; that is,

$$g_i(x, y) = f_i(x, y) + \eta(x, y) \quad (7)$$

If the previous assumptions of rapidly moving particles and a common noise for the whole set of images, then

$$\bar{g}(x, y) = \frac{1}{M} \sum_{i=1}^M g_i(x, y) \quad (8)$$

it follows that

$$E\{\bar{g}(x, y)\} = \eta(x, y) \quad (9)$$

where  $E\{\bar{g}(x, y)\}$  is the expected value of  $\bar{g}$ , and  $\bar{g}$  is the average; the next step in the procedure is to subtract the average to obtain a "clean" image. Subtracting equation 9 to equation 7, we obtain:

$$g_i(x, y) - E\{\bar{g}(x, y)\} = f_i(x, y)$$

therefore  $g_i(x, y) - E\{\bar{g}(x, y)\}$  will represent an image without the common noise of  $f_i(x, y)$  of the set.

Two major problems occur in this procedure: first, slowly moving particles can contribute to the model and thus may be erroneously eliminated from the images; second, rapidly moving particles can appear if the number of averaged images is not large enough. However, it is desirable to keep the image sequence as small as possible to minimize the computational cost and to keep the model proximate to the real background when moving average is performed.

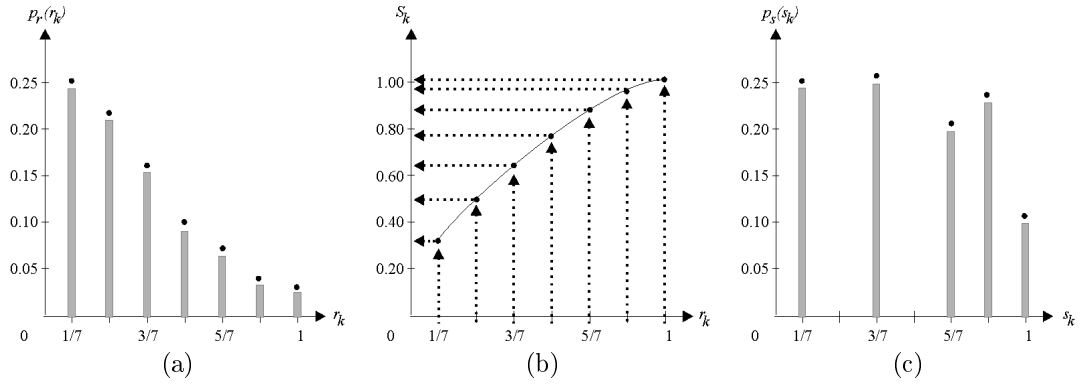
**Histogram Equalization** This image enhancement procedure uses a transformation function that alters the gray-level value of pixels so they become distributed evenly in the defined gray-scale range (0 to 255 for an 8-bit image). The function associates an equal amount of pixels per constant gray-level interval and takes full advantage of the available shades of gray.

The goal of histogram equalization is to obtain a uniform histogram. This technique can be used on a whole image or just on a part of an image. Histogram equalization will not "flatten" a histogram. It redistributes intensity distributions. If the histogram of any image has many peaks and valleys, it will still have peaks and valley after equalization, but peaks and valley will be shifted. Because of this, "spreading" is a better term than "flattening" to describe histogram equalization.

In mathematical sense, to explain histogram equalization, consider the transformation function

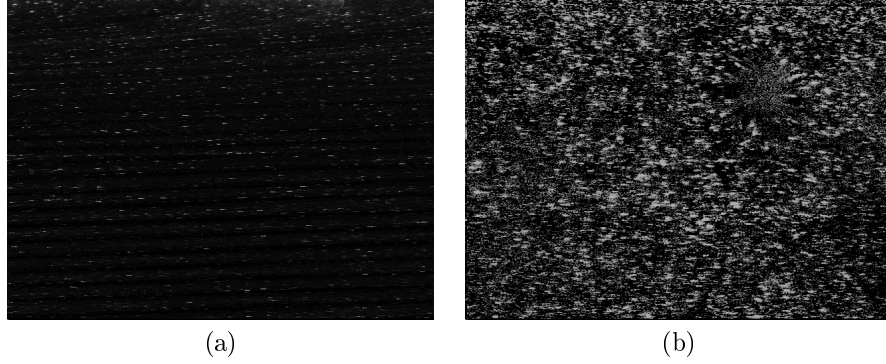
$$s = T(r) = \int_0^r p_r(w)dw \quad 0 \leq r \leq 1 \quad (10)$$

where  $w$  is a dummy variable of integration. The right side of equation 10 is recognized as the Cumulative Distribution Function (CDF) of  $r$ . The foregoing development indicates that using a transformation function equal to the cumulative distribution of  $r$  produces an image whose gray level have a uniform density. This procedure is better explained with figure 6. The original histogram of the image is shown in figure 6(a), where the transformation function to be used is shown in figure 6(b). Figure 6(c) represents the output histogram after the transformation. As can be noticed, the distribution of the histogram is more uniform, but with a reduction of the possible gray scale values within the image after histogram equalization process, the number of gray scale values can be reduced but never increased.



**Figure 6:** Illustration of the histogram-equalization method. (a) original histogram. (b) transformation function. (c) equalized histogram

The reason why histogram equalization is used in this work is that detailed information within the image will be lost when processing raw pictures because the value of gray scale level of this detailed information is sometimes below a chosen threshold. Therefore by using histogram equalization the gray scale level will be modified, showing the hidden information. As can be seen in figure 7(a) much of the information within the image is below a certain value of gray scale, but in figure 7(b) more information is available for those image analysis techniques based on gray scale thresholding.



**Figure 7:** Illustration of the histogram-equalization method. (a) original image. (b) equalized image

It's important to note that if a previous noise reduction technique is not used before histogram equalization, the noise will also increase its gray scale value, giving as a result an image impossible to analyze due to the complexity of distinguishing noise from particles.

**Gaussian Smoothing Filter** A smoothing filter attenuates the variation of light intensity in the neighborhood of a pixel. It smooths the overall shape of objects, blurs edges, and remove non desired details. Specifically, the Gaussian filter attenuates the variations of light intensity in the neighborhood of a pixel. It smooths the overall shape of objects and attenuates details. The Gaussian filter is an averaging filter; therefore the transformation function  $T$  will be the average. The *kernel* is defined with the next model

$$\begin{array}{ccc} a & d & c \\ b & x & b \\ c & d & a \end{array}$$

where  $a$ ,  $b$ ,  $c$  and  $d$  are integers, and  $x > 1$ .

Because all the coefficients in a Gaussian kernel are positive, each pixel becomes a weighted average of its neighbors. The stronger the weight of a neighboring pixel, the more influence it has on the new value of the central pixel. Unlike a smoothing kernel, the central coefficient of a Gaussian filter is greater than 1. Therefore, the original value of a pixel is multiplied by a weight greater than the weight of any of its neighbors. As a result, a greater central coefficient corresponds to a more subtle smoothing effect. A larger kernel size corresponds to a stronger smoothing effects.

One problem of using this procedure is the loss of detailed information. However it will ensure that noise will be effectively removed, and the redistribution of the gray scale of the pixels that

form a particle to having a Gaussian distribution, will help the centroid location procedure on those image analyses that search for such distribution within the particles.

**Logarithmic Gamma Correction** This image processing expands low gray-level ranges while compressing high gray-level ranges. It increases the overall brightness of an image and increases the contrast in dark areas at the expense of the contrast in bright areas. This can be explained also with figure 6. But in this case, instead of having the transformation function represented by figure 6(b), it will be a logarithmic function.

It's important to note that to have a better result from the image processing techniques, a combination of the previous discussed algorithm must be carried out. The reduction of noise by averaging is one procedure that will be always used. If not, the subsequent processes will modify the noise information constructively, making the discrimination of it harder. For example, a typical process will follow the next steps:

1. Obtain the average of the whole set of images using averaging technique.
2. Subtraction of the average image (noise) from the original images.
3. Application of the Gaussian smoothing filter.
4. Application of the logarithmic gamma correction.

**Thresholding** Thresholding is the process of discriminating pixels depending on the gray scale value. It is included just before the image analysis processes because it categorizes the pixels of the image into two main objects: pixels that belong to the background, and pixels belonging to interest objects such as particles or bubbles. The correct selection of this parameter will help separate particles that are closely together and get rid of the image inherent noise.

## IMAGE ANALYSIS

The goal of image analysis algorithms is to extract quantitative information from the image that will describe in detail some characteristics of the flow. This procedure is commonly applied just after the images have been processed with an image enhancing technique which is compatible or helpful to the analysis in question. The final result obtained at the end of this procedure is a velocity field that will represent with high accuracy the motion of the flow. This procedure can be classified in two major sub-procedures, object identification and object matching, and are different depending on the program used. The programs analyzed in this work are the home made programs “MskPTV” and “Fspots and Tracking”, and programs from outside sources “PIV Sleuth” and “AkaPTV”. These programs are described next.

### **MASKPTV PROGRAM**

MaskPTV is a PTV program for Windows, developed using VB.net. This language was chosen for its simplicity and versatility which helps in the use of modularized codes from public domain that can be downloaded from the Internet. This program arises from the fact that in the multiphase flow laboratory there is not a friendly and easy to use software capable of extracting information directly from PIV images. The development of this program is one of the objectives of this work. Therefore, it will be described in detail in the following sections.

#### **MaskPTV Object Identification and Centroid Location**

The object identification procedure classifies every pixel of the image into objects, each object having different characteristics assigned by the program user. If a pixel gray scale level is greater than the threshold, a correlation coefficient is assigned to it. This correlation coefficient is calculated using the mask cross correlation technique. Once a correlation value is assigned to each one of the pixels, they must be gathered together in groups of pixels that will form an object. Each object will store information that describes its characteristics, such as the number of pixels belonging to the object, their positions, gray scale, and correlation value. Once the objects are identified their centroids are calculated using sub-pixel interpolation. If the interpolation fails, the center of gravity method is used to locate the center with sub-pixel accuracy. The main algorithms are explained next.



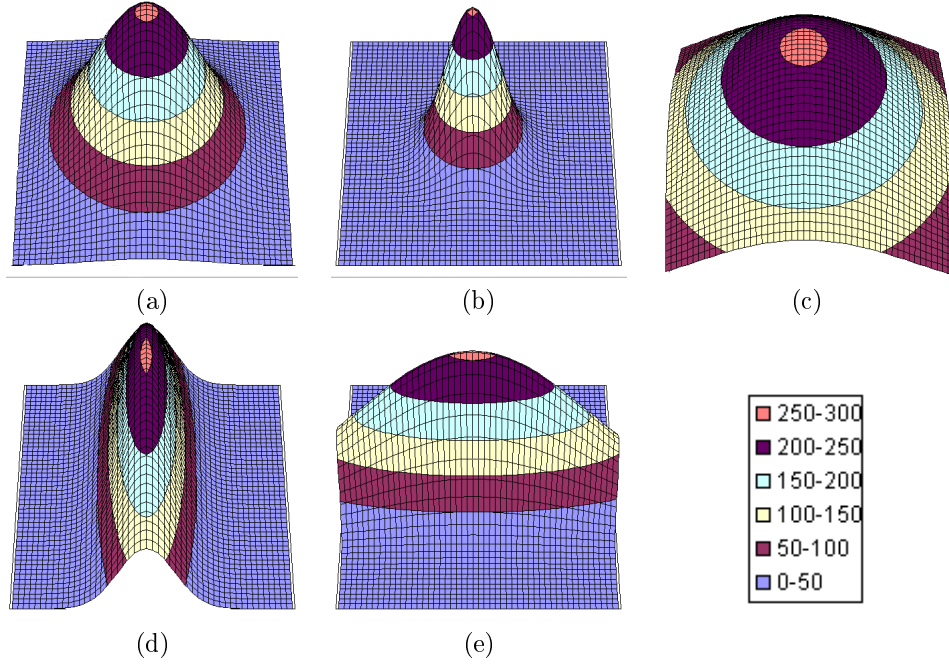
### Mask Cross Correlation Algorithm

In the MaskPTV program the mask cross correlation algorithm is used not only to locate the central pixel of the objects, but also to identify pixels that belong to the object. In other words, it can be used as a filter by using a threshold for the cross correlation value calculated. The mask cross correlation coefficient is going to be identified as  $C_1$ .

**Definition of an Object Mask** An object mask is a typical brightness pattern of a particle image. It is known as a template in pattern matching. Generally, the brightness pattern of a particle image has a peak near the center of the image, and the brightness concentrically decreases as the distance from the peak location increases. In the proposed algorithm, the particle mask is assumed to have a shape of the two-dimensional Gaussian distribution. Many particle masks have been proposed, and in this work a two-dimensional Gaussian distribution is used and is expressed as follows:

$$I(x, y) = I_0 \exp \left( -\frac{(x - x_0)^2/A^2 + (y - y_0)^2/B^2}{2r_0^2} \right) \quad (11)$$

where  $I(x, y)$  is the gray scale intensity in the position  $(x, y)$ ,  $(x_0, y_0)$  is the object centroid location,  $I_0$  is the maximum intensity of the object (positioned almost always in the center),  $A$  and  $B$  are shape modifier parameters, and  $r_0$  is the particle radius in pixels. The influence of the shape parameters on the brightness pattern is shown in figure 8, where  $r_0 = 5$ . In figure 8(a), the Gaussian shape is not modified, having a maximum gray scale value located at the center of the object. The shape modifier parameters are set both to  $A = B = 1$ . In figure 8(b), the shape modifier parameters are both set to  $A = B = 0.5$ . This is useful when the object is represented by few pixels with high gray scale value, and most of the area will have low gray scale values. In figure 8(c), both parameters are set to  $A = B = 2$ , useful when highly illuminated particles are being searched. In figure 8(d) and (e) the parameter values are  $A = 0.5$ ,  $B = 1.0$ , and  $A = 1.0$ ,  $B = 0.5$  respectively. These settings are useful when in some region of the image there exists optical deformation due to lenses or wall defects, leading to elliptical instead of circular particle images. These parameters were useful when analyzing experimental images close to the wall, where optical deformation was found.



**Figure 8:** Shape modifier parameters influence on the object mask (a)  $A=1$ ,  $B=1$ . (b)  $A=0.5$ ,  $B=0.5$ . (c)  $A=2.0$ ,  $B=2.0$ . (d)  $A=1$ ,  $B=0.5$ . (e)  $A=0.5$ ,  $B=1$

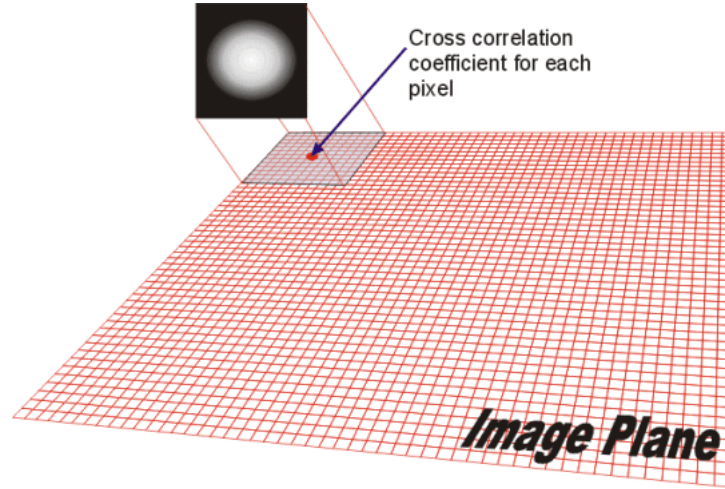
**Procedure of the Particle Mask Correlation Method** In the calculation of the particle mask correlation method, the brightness pattern of the object mask is fixed (see figure 8). The peak brightness,  $I_0$ , and the representative radius,  $r_0$ , values are fixed. The center of the object mask,  $(x_0, y_0)$ , scans all pixels in the image plane. The particle mask correlation method has the following steps.

**(1) Calculation of cross correlation coefficients**

The cross correlation coefficient between the object mask and the interrogation area in the image plane is calculated with the following equation:

$$C_1(x_0, y_0) = \frac{\sum_{i=x_0-a}^{x_0+a} \sum_{j=y_0-a}^{y_0+a} (I(i, j) - \hat{I}) (I_m(i, j) - \hat{I}_m)}{\sqrt{\sum_{i=x_0-a}^{x_0+a} \sum_{j=y_0-a}^{y_0+a} (I(i, j) - \hat{I})^2} \sqrt{\sum_{i=x_0-a}^{x_0+a} \sum_{j=y_0-a}^{y_0+a} (I_m(i, j) - \hat{I}_m)^2}} \quad (12)$$

Here  $C_1(x_0, y_0)$  is the cross correlation coefficient at  $(x_0, y_0)$ ,  $I(i, j)$  is the brightness value of particle image plane at  $(i, j)$ , and  $I_m(i, j)$  is the brightness value of the particle mask at  $(i, j)$ . The interrogation area,  $a \times a$ , is chosen by the user.  $\hat{I}$  and  $\hat{I}_m$  are brightness averages of the interrogation area and the object mask image, respectively. The particle mask scans all pixels on



**Figure 9:** Cross correlation coefficient plane

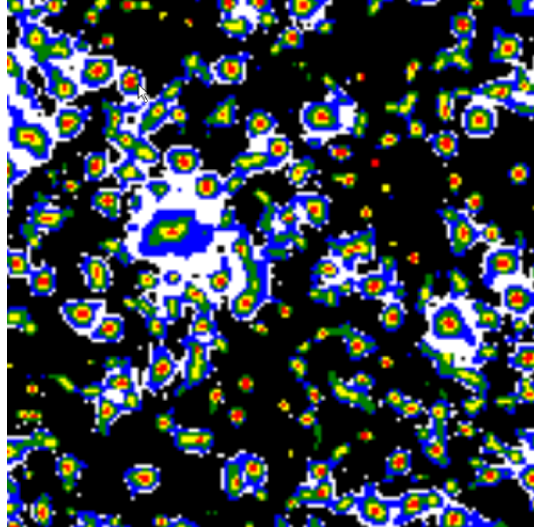
the image plane, which produces a correlation coefficient plane. This is better explained in figure 9. Every pixel receives a cross correlation coefficient value, from -1 to 1, depending on how closely the pixel in the  $(i, j)$  location represents the center of a Gaussian distribution, which is similar to the center of the object mask.

## (2) Extraction of object area

Two thresholds are considered in this procedure. The gray scale threshold, (GST), and mask cross correlation coefficient threshold, ( $C_1T$ ). Pixels having gray scale values and correlation coefficients larger than these thresholds are candidates to be part of an object. The values of these thresholds will depend on the application. If the researcher is interested only on the centroid location of objects, the only necessary threshold is the  $C_1T$ . On the other hand, if the researcher is interested in the size of the objects, a good GST value must be selected. An example of the application where both thresholds are used is shown in figure 10, where different colors are used to represent the  $C_1$  value of the pixels, where blue pixels range from 0.0 to 0.2, green from 0.2 to 0.4, yellow from 0.4 to 0.6, and red from 0.6 to 1.0. White color represents pixels with a higher gray scale value than the GST, but a  $C_1$  value lower than 0.0. Black color represents pixels with a gray scale and  $C_1$  values lower than the thresholds.

When using both thresholds in the same analysis, a pixel will be considered as part of an object only if its values are larger than the thresholds, after first using the GST as a filter. Pixels with lower gray scale values are not considered and will receive automatically a  $C_1$  value of -1.0. This helps because the mask cross correlation technique is inherently slow, therefore reducing the number

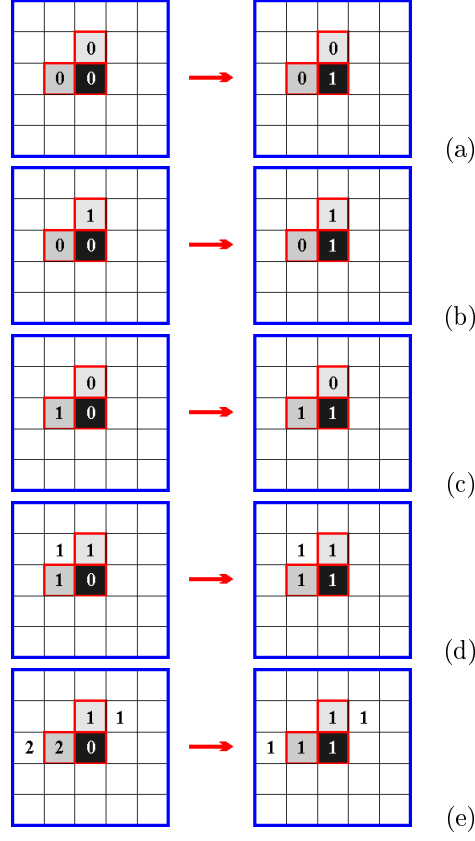
of pixels needing to be analyzed will reduce the computational costs. Once the pixel passes through the GST filtering successfully, the cross correlation coefficient is calculated for that particular pixel, by using the calculation all its neighbour pixels inside the interrogation area.



**Figure 10:** Thresholding influence in object area detection. Colors white = pixels with a gray scale value larger than GST and  $C_1$  lower than 0.0, blue = pixels with  $C_1$  value from 0.0 to 0.2, green = pixels with  $C_1$  value from 0.2 to 0.4, yellow = pixels with  $C_1$  value from 0.4 to 0.6 and red = pixels with  $C_1$  value from 0.6 to 1.0

After a pixel is classified to be part of an object, the next step is to determine to which object this pixel belongs, this procedure is done with the top-left algorithm. By making a sweep from left to right and top to bottom a identification number is assigned to each pixel as follows:

1. A pixel whose left and top neighbours does not belong to an object becomes the first pixel of a new object (figure 11(a)).
2. A pixel whose top neighbour belongs to an object and the left neighbor does not belong to any object becomes part of the top neighbour object (figure 11(b)).
3. A pixel whose left neighbor belongs to an object and the top neighbor does not belong to any object becomes part of the left neighbour object (figure 11(c)).
4. A pixel whose left and top neighbours belong to the same object becomes part of that object (figure 11(d)).
5. A pixel whose top and left neighbours belong to different objects becomes part of the top neighbour. Also, all pixels that belong to the same object as the left neighbour will be reassigned as part of the top neighbour object (figure 11(e)).



**Figure 11:** Top-left assignment algorithm, the numbers represent the object identification

### (3) Estimation of object centroid location

It is well known that most of the PIV error derives from the object centroid location. The mask cross correlation technique gives only the pixel position where the object centroid is located. In other words, it uses integer pixel units, with an average error within 0.5 pixels for the centroid's location. In order to reduce the error, the Gaussian three point sub-pixel interpolation method and center of gravity method are used. These methods have sub-pixel accuracy; therefore, the final result of this procedure is the object centroid pixel position plus or minus a sub-pixel correction, which is explained in the next paragraph.

Consider  $(x_0, y_0)$  to be the pixel position with a higher  $C_1$  value within an object. Determine with sub-pixel accuracy the centroid location of the object by means of adding the sub-pixel correction  $\epsilon_x$  for the  $x$  coordinate and the sub-pixel correction  $\epsilon_y$  for the  $y$  coordinate. Using the Gaussian three point sub-pixel interpolation, these corrections are expressed with

$$\epsilon_x = \frac{\log(I(x_0 - 1, y_0)) - \log(I(x_0 + 1, y_0))}{2 \log(I(x_0 - 1, y_0)) + \log(I(x_0 + 1, y_0)) - 2 \log(I(x_0, y_0))} \quad (13)$$

and

$$\epsilon_y = \frac{\log(I(x_0, y_0 - 1)) - \log(I(x_0, y_0 + 1))}{2 \log(I(x_0, y_0 - 1)) + \log(I(x_0, y_0)) - 2 \log(I(x_0, y_0 + 1))} \quad (14)$$

The particle centroid position with sub-pixel accuracy  $(x_c, y_c)$  will be calculated using:

$$x_c = x_0 + \epsilon_x \quad (15)$$

and

$$y_c = y_0 + \epsilon_y \quad (16)$$

Some reasons that cause the three point sub-pixel interpolation to fail are:

- Saturation of pixels close to the center of the object, i.e. the pixels used for the interpolation have the same gray scale value.
- The pixel with highest  $C_1$  value is not the one with the largest gray scale value between the three points needed for the interpolation.

When the calculated sub-pixel corrections  $\epsilon_x$  or  $\epsilon_y$  are greater than 1.0 or smaller than -1.0, the sub-pixel interpolation is considered a failure, and the center of gravity method is used instead. This method is expressed with:

$$\epsilon_x = \frac{\sum_{k=1}^K x_k I(x_k, y_k)}{\sum_{k=1}^K I(x_k, y_k)} \quad (17)$$

and

$$\epsilon_y = \frac{\sum_{k=1}^K y_k I(x_k, y_k)}{\sum_{k=1}^K I(x_k, y_k)} \quad (18)$$

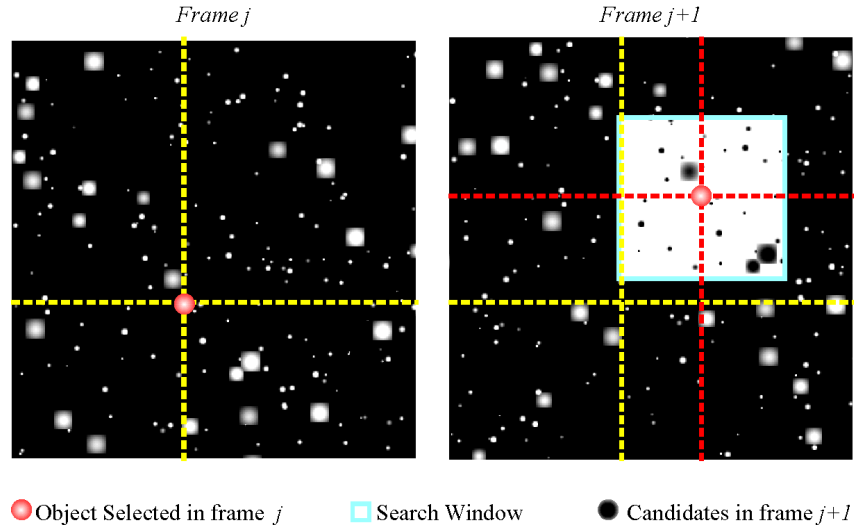
where  $K$  is the total numbers of pixels belonging to the object.

### MaskPTV Object Matching or Particle Tracking

After objects represented in the PIV pictures have been correctly located, they are labeled following a different nomenclature where,  $O_{i,j}(x_j, y_j)$  represents object  $i$  with position  $(x_j, y_j)$  in time interval

or picture frame  $j$ . For example, the object  $O_{1,1}(35, 150)$  represents object 1 with center position  $x_1 = 35, y_1 = 150$ , in frame or time interval 1. In this work two-frame single exposure pictures were used. This means that a given object will be represented in two different frames, frame  $j$  with label  $O_{i,j}(x_j, y_j)$ , and in frame  $j + 1$ , with label  $O_{i,j+1}(x_{j+1}, y_{j+1})$ . The procedure which relates the representations of the same object in different frames is described in the following sentences.

**Candidate Window** Considering all objects found in frame  $j + 1$  as candidates to be a pair of objects in frame  $j$ , is a tedious and slow process. Therefore, if possible, some previous knowledge of the flow is used to reduce the searching area to the most probable region where this object could have been moved in the time interval  $\Delta t_{PIV}$ . This region is called the candidate window. The area and position of the candidate window depends on the flow characteristics. All objects inside this candidate window will be considered as pair candidates for the selected object in frame  $j$ . In figure 12 the red object is represented in frame  $j$  with initial position  $(x_j, y_j)$  (yellow lines), and in frame  $j + 1$  with final position  $(x_{j+1}, y_{j+1})$  (red lines); the candidate window is represented by the blue square, and pair candidate objects are represented inside the search window as black objects.



**Figure 12:** Search window

An important characteristic of the candidate window in MaskPTV program is its flexibility. Size and position of the rectangular area can be chosen as follows:

Candidate window right size	$CW_r$
Candidate window left size	$CW_l$
Candidate window up size	$CW_u$
Candidate window down size	$CW_d$
Candidate window initial $x$ position	$CW_{x_i} = x_j - CW_l$
Candidate window final $x$ Position	$CW_{x_f} = x_j + CW_r$
Candidate window initial $y$ Position	$CW_{y_i} = y_j - CW_u$
Candidate window final $y$ Position	$CW_{y_f} = y_j + CW_d$

It is important to note that the previous parameters for the candidate window depends entirely on the user experience. If a candidate window area is too big, there will be a larger probability of finding spurious vectors because of the excess of candidates. If the candidate window area is too small, the selection of candidates will be erroneous because the correct object will not be considered as a candidate.

**Tracking Cross Correlation Coefficient or Tracking Coefficient** The tracking coefficient (TC) is a measure of how closely related, or how similar an object in frame  $j$  is to an object in frame  $j+1$ . The pairing procedure using the TC is as follows: First, an object located in frame  $j$  is selected. This object will have as many candidates as there are objects within the candidate window in frame  $j+1$ . Second the TC value between each of the candidates and the selected object is calculated. Finally, the candidate that possesses the largest TC is considered to be the most probable pair for the selected object in frame  $j+1$ . The previous procedure is repeated for all the objects in frame  $j$ . The main difference between obtaining the TC by using MaskPTV and the former binarization tracking routine is the use of all the information within the image by MaskPTV program (all the 256 gray-scale levels), while in the binarization tracking routine, a binarized image (black and white) resulting from the gray scale thresholding is used for the calculation. Consequently, a lot of information is lost.

To obtain the TC, two windows must be defined per each calculation, one at the position surrounding the selected object in frame  $j$ , and the other surrounding the selected candidate object in frame  $j+1$ . Both windows must have the same area and geometry. This windows are called the tracking coefficient windows (TCW). For each object in frame  $j$  it will exist  $nc+1$  TCW's, where  $nc$  is the number of candidates in frame  $j+1$  corresponding to that selected object. In a typical experimental image the average number of objects ( $no$ ) found is around 4500, each having on average 7 candidates. The number of TCW's in this case is  $(nc+1)(no) = 8 \times 4500 = 36000$



and the number of TC's calculated is  $(nc)(no) = 7 \times 4500 = 31500$  TC's. This example is for medium density PIV images. Working with higher density images the TC calculations increase exponentially, making this process very expensive computationally.

The selection of the shape and size of the TCW is also dependent on the user experience, but normally the TCW is smaller than the candidate window, in figure 13, ten TCW's are shown. The first one (bottom left of figure 13) determines the gray-scale pattern to be correlated, with the characteristic of being centered on the object selected in frame  $j$ . The other nine windows are centered on each one of the objects inside the candidate window in frame  $j + 1$ , and every candidate window is correlated with the pattern window. The correlation coefficient obtained from this calculation is shown on the right side of figure 13. Candidate number five is the one with a higher correlation coefficient, therefore it is concluded that the selected object in frame  $j$  is the same object as candidate five, but in a different position and time.

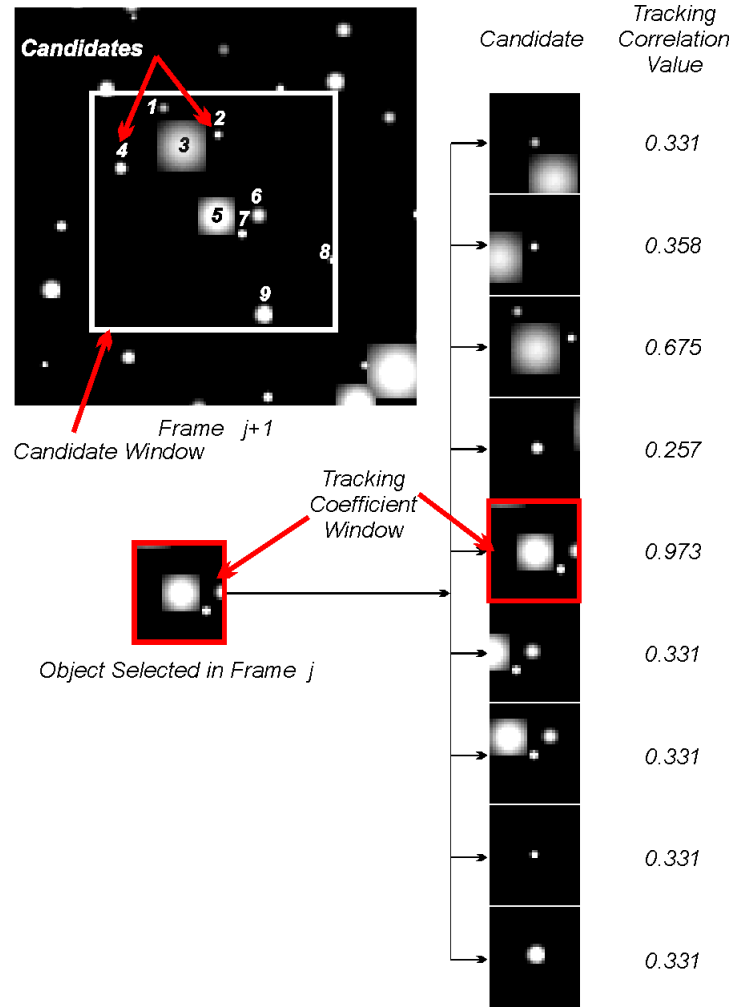
The method used to calculate the tracking coefficient is the mask cross correlation technique. In this case, instead of having a single image pattern like a Gaussian distribution used on the particle centroid location algorithm, there will exist as many image patterns as objects found in frame  $j$ . We can call this a dynamic mask cross correlation.

**Iterative Pairing Procedure** In this procedure, each particle in frame  $j$  have selected their probable particle candidates in frame  $j + 1$ , and have a TC for each candidate; the selection of the most probable candidate is easy -it is the one with the largest TC. Most of the time the TCW of an object overlaps with other objects TCW's, therefore some objects in frame  $j + 1$  are pair candidates for more than one object in frame  $j$ . To avoid spurious pairing an iterative checking is used. If a candidate is the "best candidate" for two objects, it will be assigned to the object with higher TC and will be erased from the list of candidates of the other object, allowing the next candidate from the new list to become its "best candidate".

**Velocity Field Calculation** After the pairing of objects is successful, the position of all the objects in frame  $j$  and in frame  $j + 1$  is known. Recalling that the time interval between these frames is also known, the velocity components are computed with:

$$u_i = \frac{dx_i}{dt} = \frac{x_{i_2} - x_{i_1}}{\Delta PIV} \quad (19)$$

$$v_i = \frac{dy_i}{dt} = \frac{y_{i_2} - y_{i_1}}{\Delta PIV} \quad (20)$$



**Figure 13:** Example of tracking coefficient windows (TCW) and object matching procedure

where  $u_i$  and  $v_i$  represent the velocities in the stream-wise and normal flow directions of object  $i$ ,  $x_{i_1}$  and  $x_{i_2}$  represent the  $x$  coordinate position of object  $i$  in frame  $j$ , and  $j+1$  respectively, and  $y_{i_1}$  and  $y_{i_2}$  represent the  $y$  coordinate positions of object  $i$  in frame  $j$  and  $j+1$ .

## FSPOTS & TRACKING PROGRAM

This program was developed in Fortran 77, and it runs in Unix system. To find the centroid location of the objects, the subprogram Fspots employs the center of gravity method described before (equation 17), and for the calculation of the particle displacement the subprogram Tracking uses the binarization cross correlation method (Hassan et al 1992).

## AKAPTV PROGRAM

This software for Windows was developed by Dr. Tomomasa Uemura (Uemura and Yamamoto 1993), and is a PTV based algorithm based on the similarity of particle distribution patterns. In order to identify each particle motion in a short time interval, a cross correlation method is employed after the pictures of particle images are binarized. The merits of the technique are (1) simple and very high speed algorithm for particle tracking, (2) computer memory size required is small, and (3) the image analysis system is cheap and simple. AkaPTV program uses the center of gravity technique for the centroid locations of objects and the binary cross correlation method to track particles.

## RESULTS AND DISCUSSION

The results section will be divided in four main parts. The first part will cover the algorithms validation of the MaskPTV program using the home made artificial images. The second part will deal with the comparison of the different programs analyzed in this study by using the PIV challenge standard images. In the third part, experimental results are presented as examples of the use of these programs. The last part of the results section will compare the influence of different image processing techniques over the results of analyzing experimental images.

### HOME MADE ARTIFICIAL IMAGES

The home made artificial images were created in order to have a way to completely validate the algorithms of the MaskPTV program. A sensitivity analysis was developed and is explained in the next sections.

#### Different Object Radius

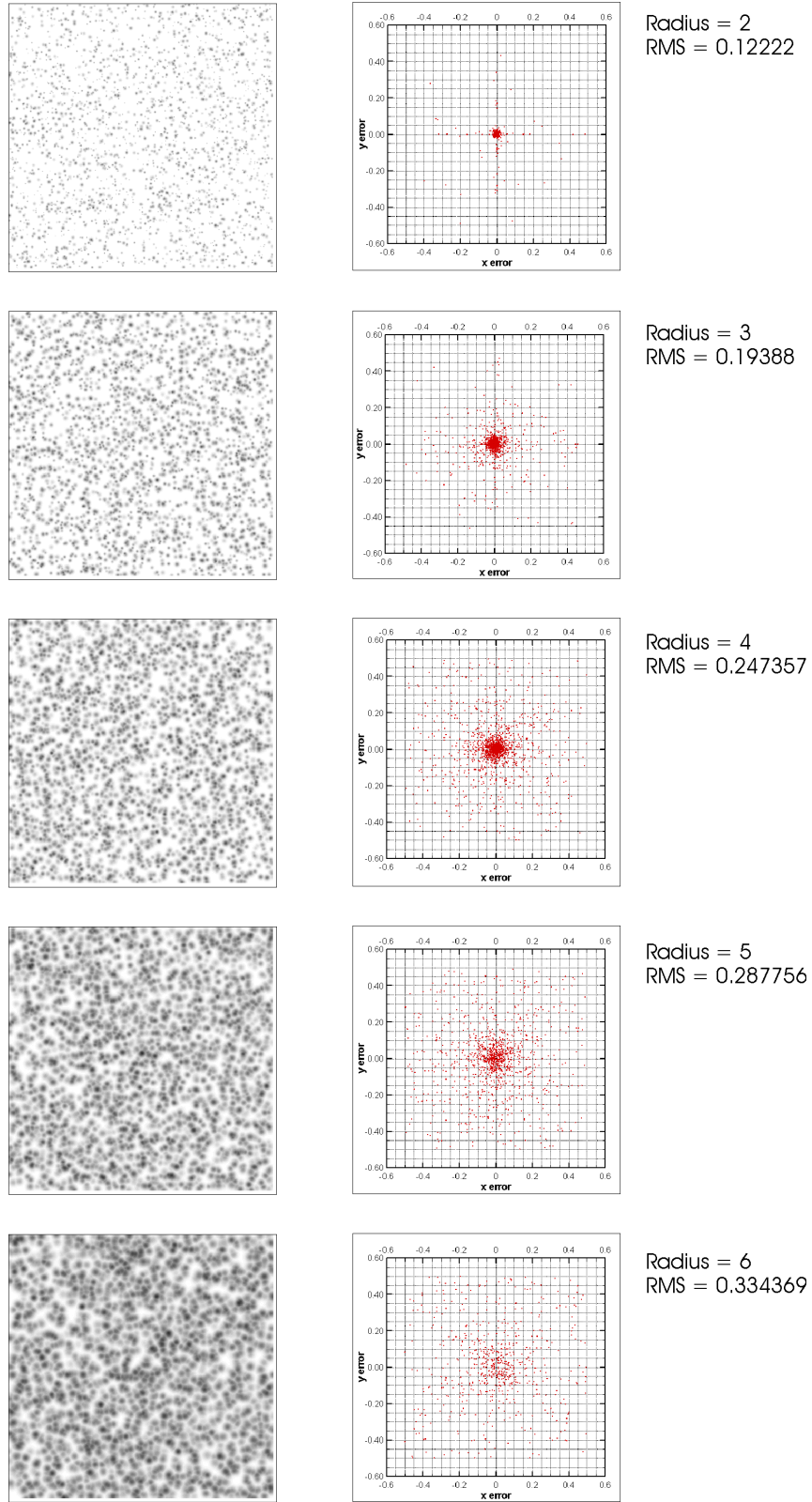
Images with different object sizes were generated in order to see how the accuracy of the particle centroid location algorithm changes. The radii of the objects considered were 1, 2, 3, 4, and 5 pixels. Samples of the images and the results are shown in figure 14. The figures on the right of the artificial images represent the displacement errors  $(\Delta x, \Delta y) = (x_{real} - x_{calc}, y_{real} - y_{calc})$ , and the root mean square value for centroid location (RMS) is also shown. The RMS is defined by:

$$\text{RMS} = \frac{1}{K} \sum_{k=1}^K \sqrt{(x_{real} - x_{calc})^2 + (y_{real} - y_{calc})^2} \quad (21)$$

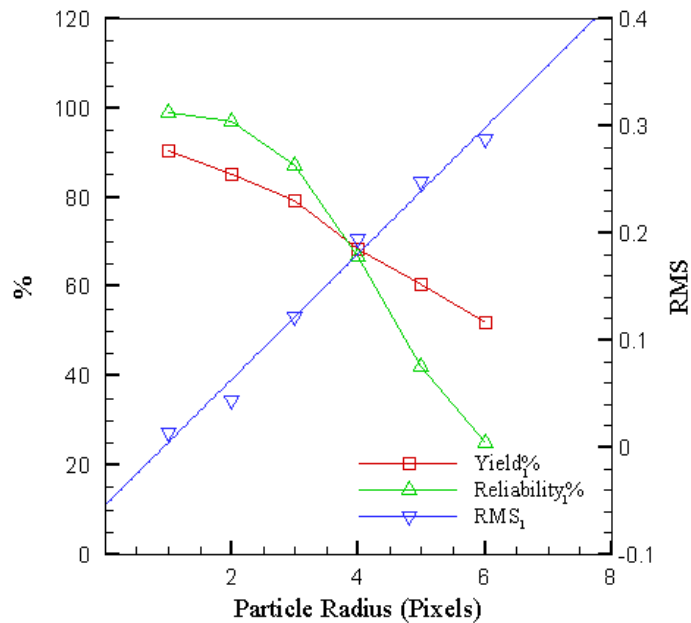
More information is displayed in figure 15 where the yield, reliability, and RMS of objects located are functions of the particle radius. The definition of yield and reliability are:

$$\text{yield} = \frac{\text{number of objects located}}{\text{total number of objects}} \quad (22)$$

$$\text{reliability} = \frac{\text{number of objects correctly located}}{\text{total number of objects located}} \quad (23)$$



**Figure 14:** Different particle radius images and related RMS error

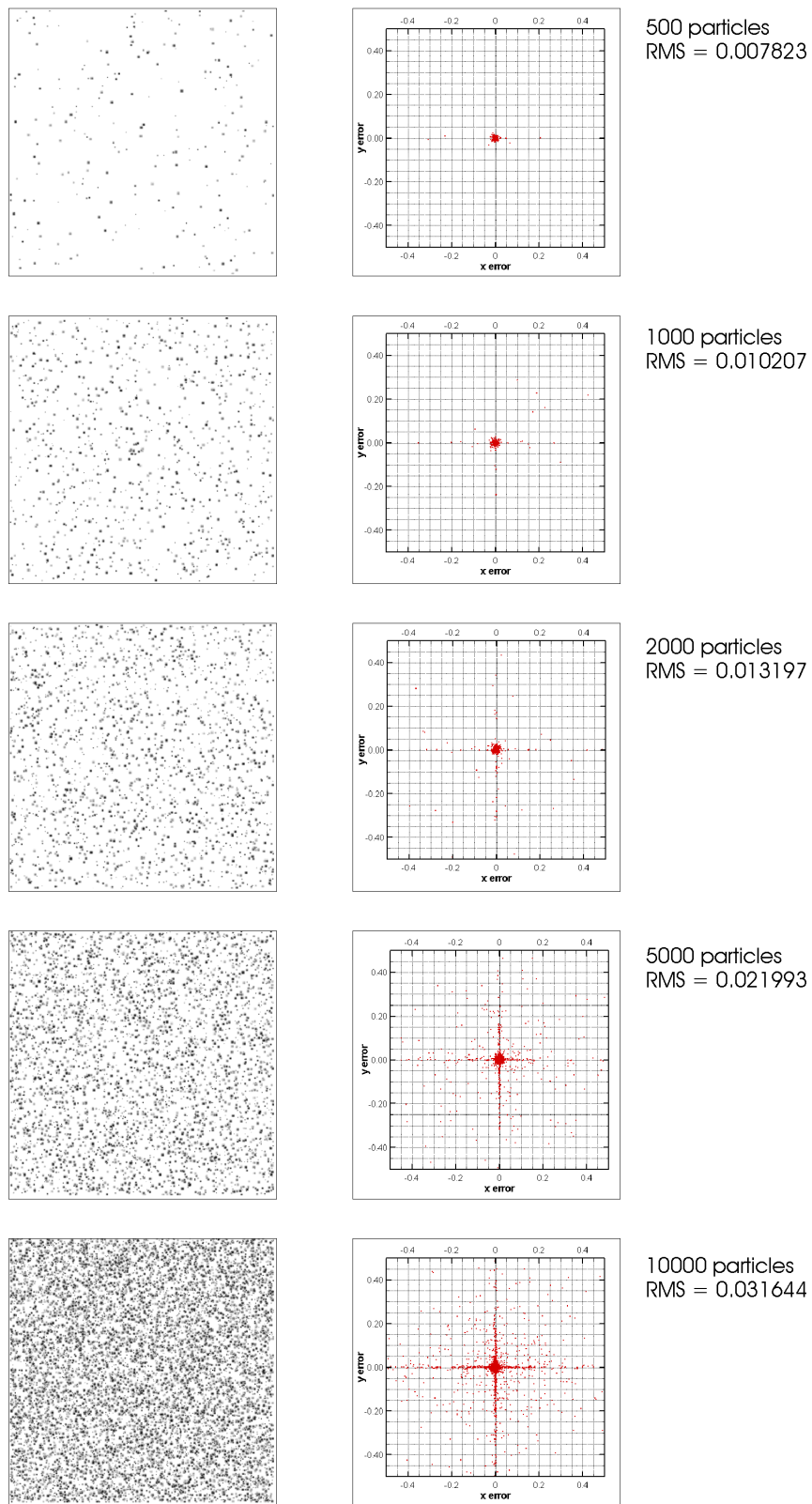


**Figure 15:** Yield, reliability, and RMS as a function of object radius

As expected, the lower the object radius, the better, in general, are the results of the analysis. This is due to several reasons. First, the concentration of objects was kept the same for all cases. As can be seen from figure 14, big objects overlap each other making it virtually impossible to discern individual particles from the chunks shown in the picture. What MaskPTV does is obtain a centroid location for a group of closely positioned objects, and that is why the RMS increases so much. Also, the reliability decreases because less particles are identified even if there exists the same quantity of particles in all sets.

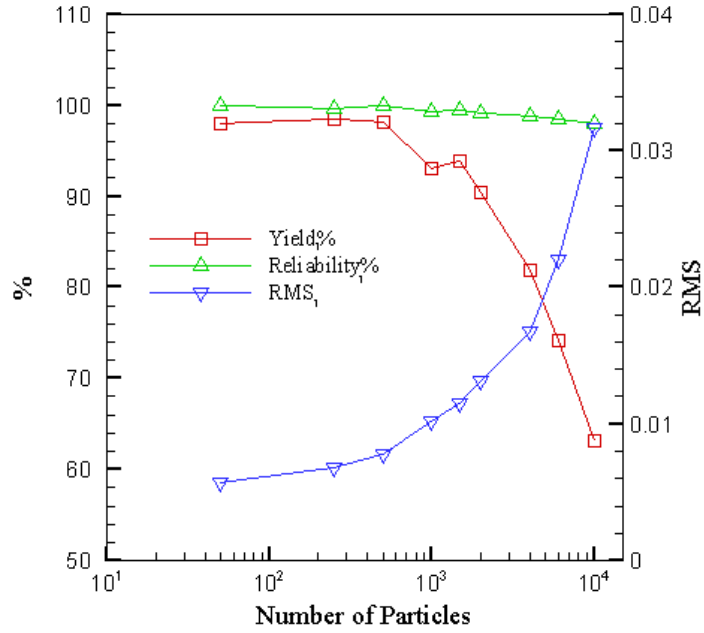
### Different Concentrations

A radius value of 1 pixel was selected to ensure little overlapping over a wide range of concentrations. Different cases where the concentration of objects varies from 500 objects per image to 10000 objects per image were considered. An example of these artificial images and its related displacement errors are shown in figure 16.



**Figure 16:** Different concentration images and the resulting centroid error

The plots of the centroid deviations in figure 16 indicate there is a small peak locking. This is expected to happen with PTV when using objects that are too small. The sub-pixel interpolation and the center of gravity algorithms allow the finding of centroid locations with high accuracy. From figure 17, it's noticeable that the yield decreases with the concentration; this result is normal because, as mentioned before, there exists object overlapping. The RMS also increases with the concentration, but only until a maximum value of 0.003 which is still very low.

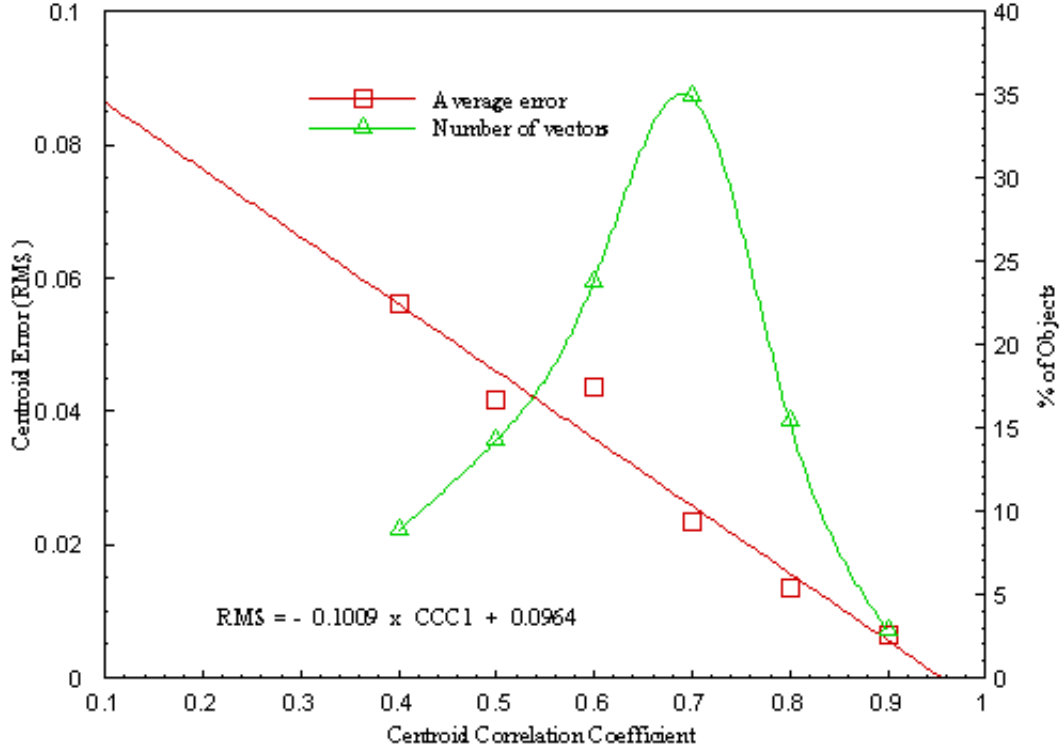


**Figure 17:** Yield, reliability and RMS of object centroid estimation as a function of object concentration

In order to gain more information in figure 18 plots the population of objects and RMS as a function of  $C_1$ . A fitted equation of RMS was obtained to be used later in order to estimate an RMS value for the centroid location of objects with real images. This is possible because on the analysis of any image (experimental or artificial) one of the output values is the value of the  $C_1$  for each object detected.

From figure 18 it can be concluded that most of the objects are detected with a  $C_1$  value from 0.6 to 0.8, The higher the correlation value, the lower the error.





**Figure 18:** Population of objects and RMS depending on the  $C_1$

### Different Noise Percentage

Choosing a fixed value for the number of particles per image (1500), a Gaussian noise was introduced ranging from 0 to 100 percent. The resulting RMS for the centroid location as a function of  $C_1$  for different noise percentage is shown in figure 19, where the maximum error obtained was 0.4 for a noise percentage of 100%. Again, the results from figure 19 will help relate the errors with parameters commonly obtained on the analysis of both artificial and real images. From this analysis five more curves were obtained that can be used later with artificial images in similar noisy situations.

Figure 20 shows the population plots vs  $C_1$ , and it's clear that most of the objects found in the analysis report a  $C_1$  value between 0.6 to 0.7 for the cases with a noise percentage greater than zero.

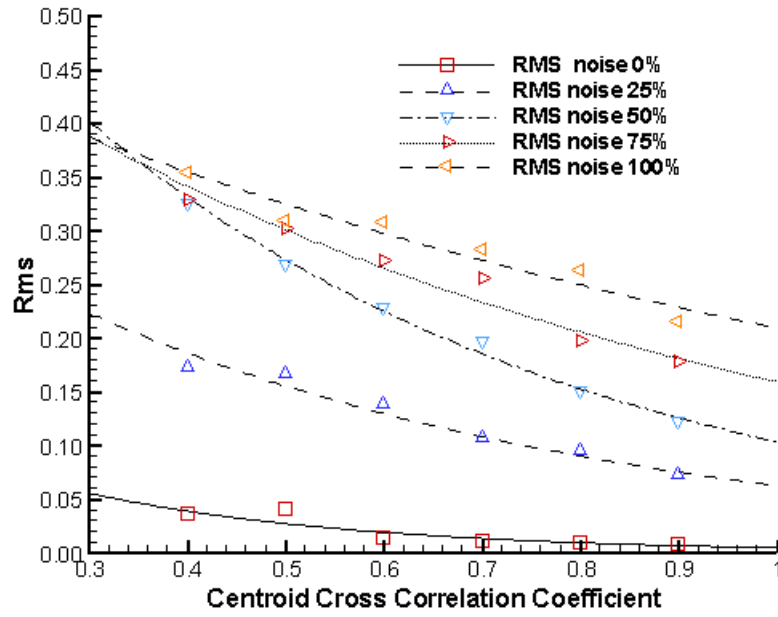


Figure 19: RMS depending on the  $C_1$  and noise percentage

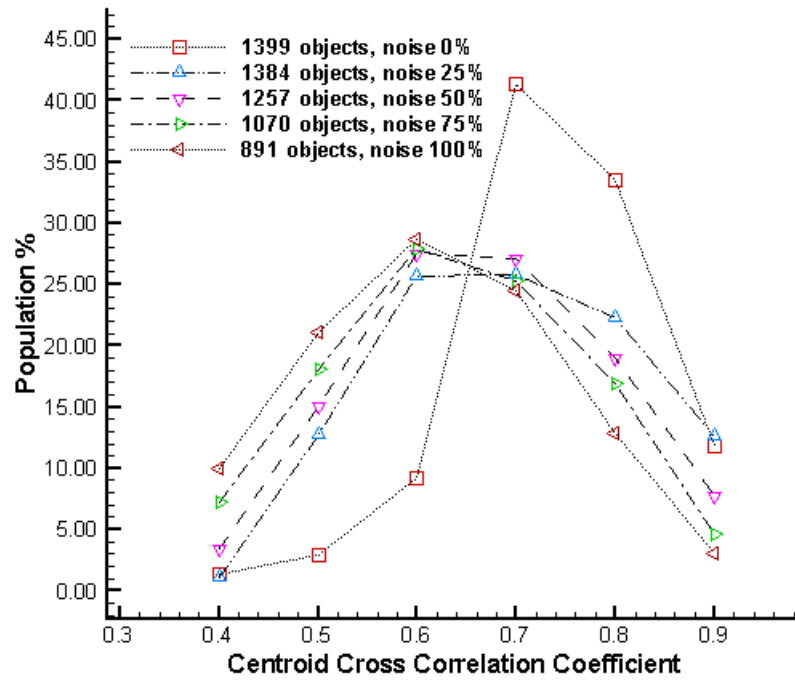


Figure 20: Population depending on  $C_1$  and noise percentage

To obtain an estimate of the error on the final vector location, the RMS, yield, and reliability are re-defined:

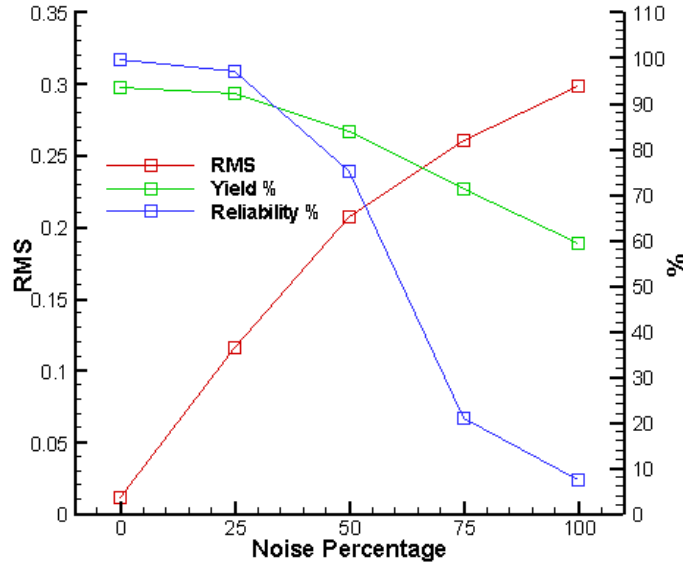
$$\text{RMS} = \frac{1}{K} \sum_{k=1}^K \sqrt{(u_{real} - u_{calc})^2 + (v_{real} - v_{calc})^2} \quad (24)$$

$$\text{yield} = \frac{\text{number of vectors located}}{\text{total number of vectors}} \quad (25)$$

$$\text{reliability} = \frac{\text{number of vectors correctly located}}{\text{total number of vectors located}} \quad (26)$$

Note that equations (21), (22) and (23) are different.

The yield, RMS, and reliability plots for vectors are shown in figure 21, where the RMS increases from almost zero for 0% noise to a value of 0.3 for 100% noise. Reliability also decreases from 95% to a 10%. This is due to the large quantity of spurious vectors found when the noise was set to 100%.



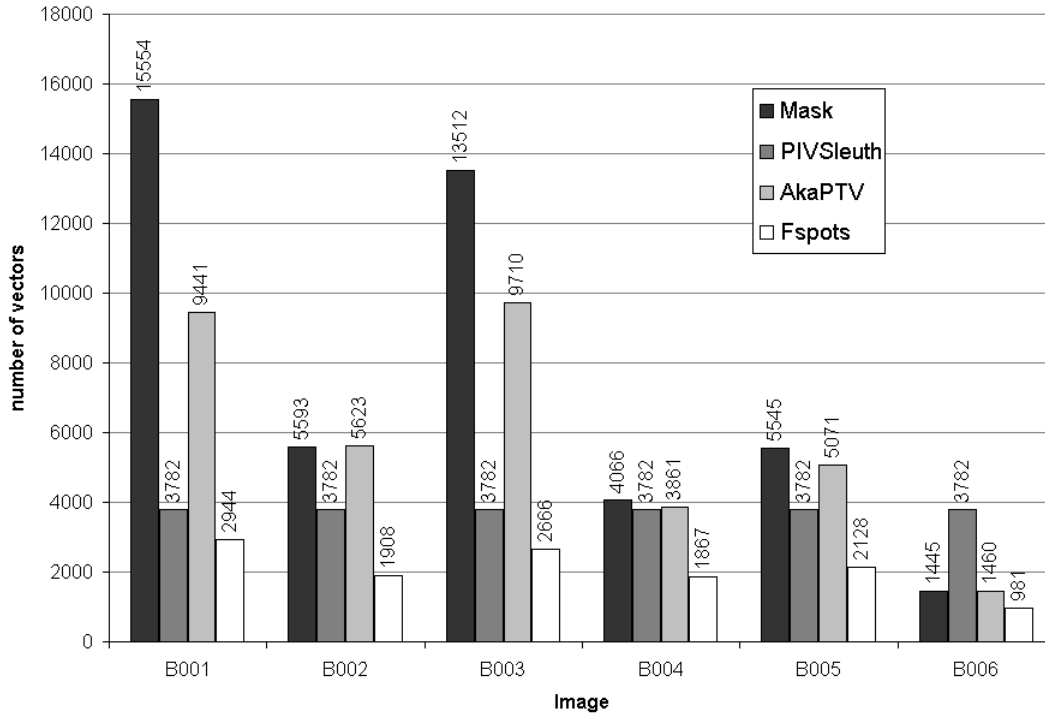
**Figure 21:** RMS, yield, and reliability of vectors location estimation for different noise percentage

To make more clear the applicability of figure 21 to real cases, assume that the estimated noise percentage of a real experimental image is similar to a 20% Gaussian noise. From the RMS curve, the value of the average error of the velocity field is 0.1. A 20% noise is elevated considering that it can theoretically be removed by means of the average image processing.

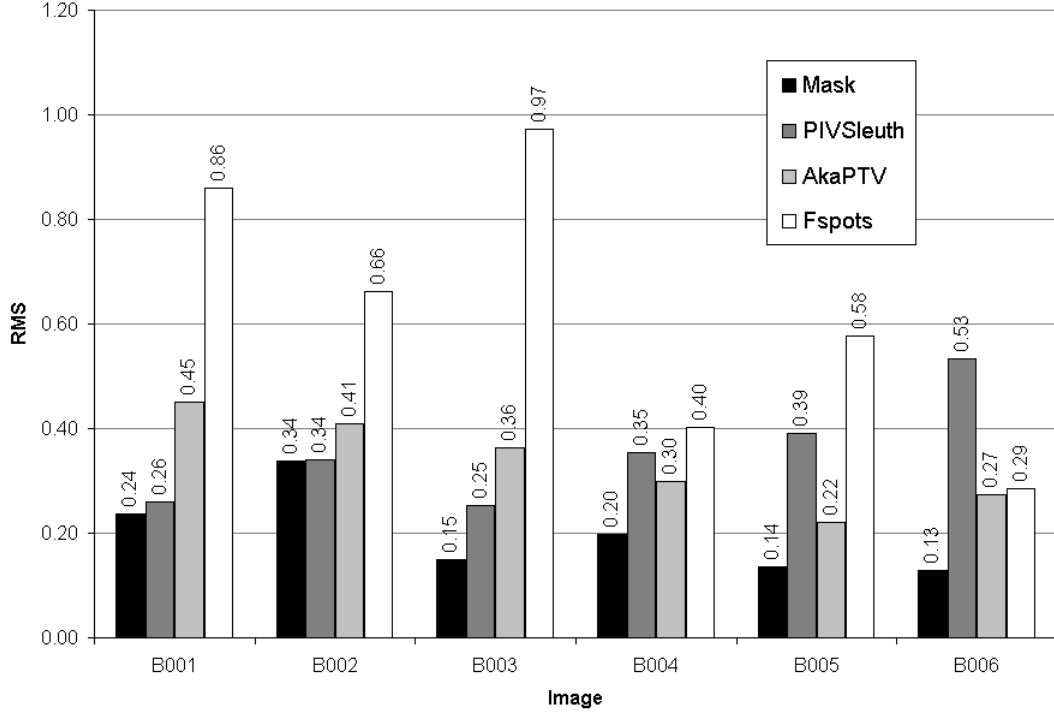
## PIV CHALLENGE RESULTS

As explained before, the images used in this section are the same images provided by the first international PIV challenge. The images were not modified by any image enhancing technique because they were generated without noise source. Any pixel with a gray level intensity greater than zero is considered to be part of a particle. The image analysis procedure was performed with four different programs: MaskPTV, PIVSleuth, AkaPTV, and FspotsPTV. The details of the results of each of these programs are given next:

Figure 22 summarizes the number of vectors found by the image analysis programs for each of the cases considered, recalling that there are small particle images (B001, B003, B005) and large particle images (B002, B004, B006) with different concentrations. Figure 23 shows the RMS calculated for the same images. It clearly shows the influence of the algorithm, the particle concentration, and the diameter on both the number of detected vectors and in the error.



**Figure 22:** Number of vectors found by different image analysis techniques using PIV challenge images



**Figure 23:** RMS of vectors located by different image analysis techniques using PIV challenge images

To clarify the differences observed, the cumulative histogram is used and it's given by:

$$p(\varepsilon) = \frac{\int_0^\varepsilon p(e)de}{\int_0^\infty p(e)de}$$

where

$p(\epsilon)$  is the population of vectors having an error lower than  $\epsilon$  and

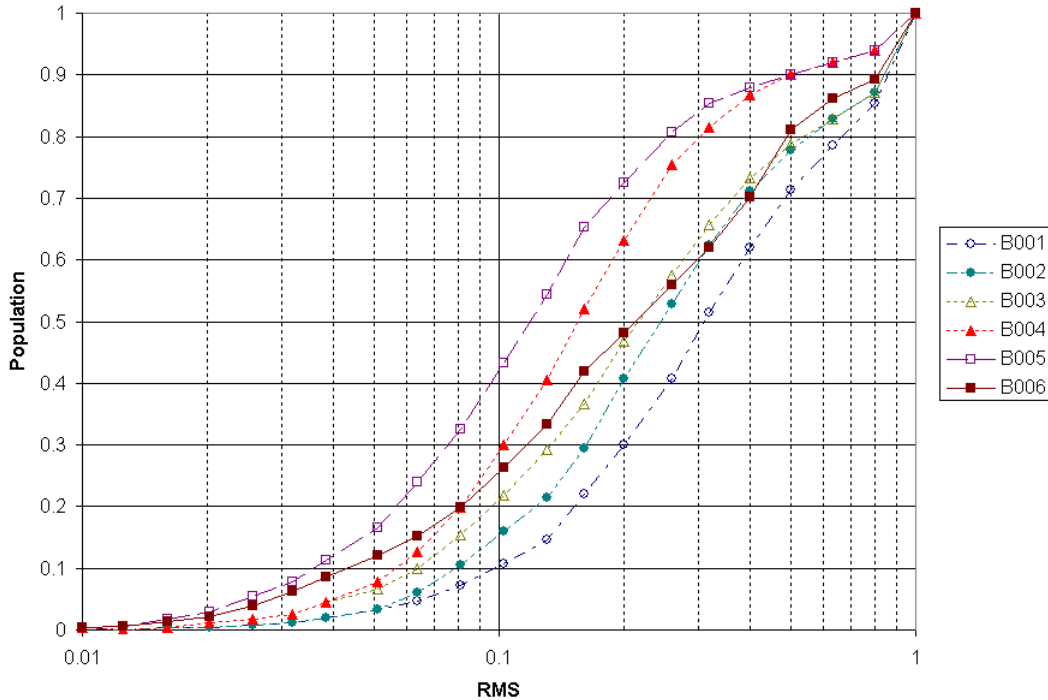
$\epsilon$  is the difference between the real displacement value and the calculated one.

The cumulative histogram or probability density function is a good representation of the accuracy of the vectors found. Lines which approach the top left corner indicate a good accuracy, and the ones going toward the bottom right indicate a bad accuracy.

A comparison of the influence of the different cases (B001-B006) over the accuracy of each algorithm is given. Also presented are figures where the accuracy of the algorithms for each case is compared. From the previous figures MaskPTV clearly shows the best results are obtained with low concentration almost independently of the particle size, but that medium concentration does fairly well too.

### PIVSleuth Results

In this program, the iterative interrogation technique was used, which follows three steps: First, a preliminary analysis was done using an initial FFT size of 128 pixels, and an initial and final search window area of 64 pixels. Second, The results of the first analysis was used as a base for the second iterative interrogation, using a FFT of 64 pixels and an initial and final search window area of 32 pixels. Third, the results of the second analysis were used as a base for the third interrogation, using smaller window sizes of 32 pixels for the FFT, and 16 pixels for the initial and final search window. A window overlapping of 50% and a search mean particle diameter of 3 pixels were used for the three steps. This procedure was done to increase the amount and accuracy of the velocity field obtained. Figure 24 shows the population of the error for the PIVSleuth program. The best results are obtained with the low density-small particles (case B005). This result is unusual because PIV techniques are supposed to behave better with the high concentration-big particles (case B002). The erratic behaviour of these results may be due to the fact that the same initial velocity field was used as a base in the iterative interrogation for all the different cases.

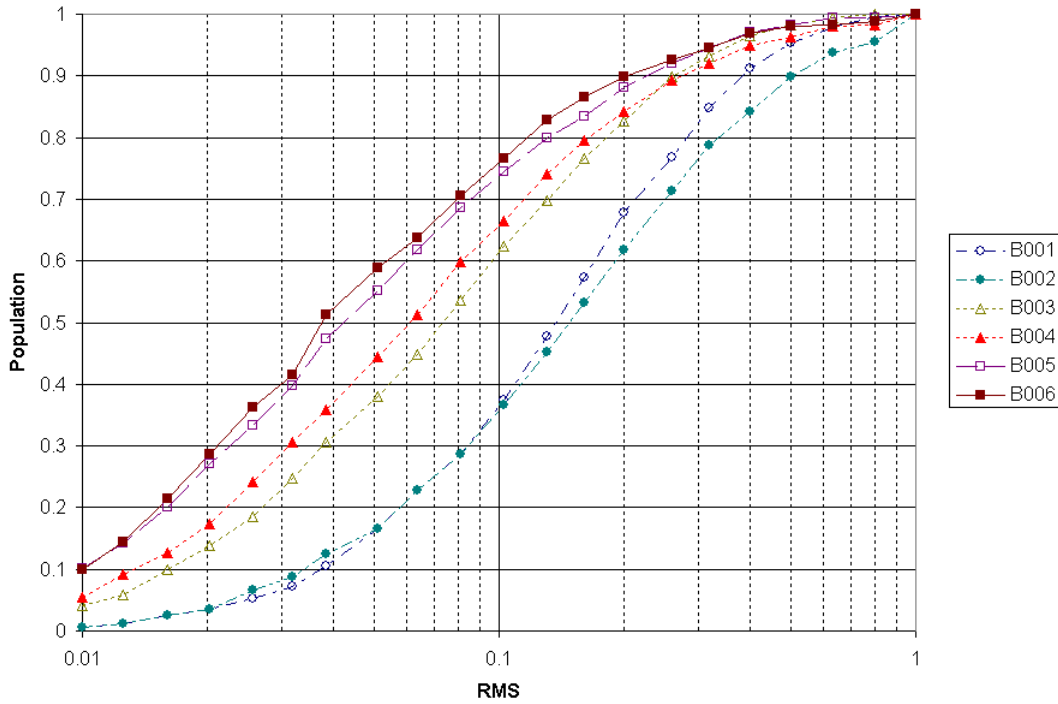


**Figure 24:** RMS probability density function of the displacement for different image concentration and sizes. Analysis with PIVSleuth

## MaskPTV Results

In this program, the analysis by regions feature was used, dividing the image into 5 rectangles, each rectangle representing one region of the image where different and independent parameters can be set. A higher searching area was set near the vortex center located in the center region (Region 5), and a smaller searching area in the surrounding regions (Regions from 1 to 4). To help increase the number of vectors found, the cross correlation threshold for tracking particles has a lower value near the vortex center. There exists a higher velocity gradient that makes the matching of particles more difficult due to the low correlation between frames. For the same reason the cross correlation window was smaller in the vortex center with an area of 8 x 8 pixels in contrast with an area of 12 x 12 pixels in the surrounding regions.

The parameters for detecting particle centroids depended only on the particle size. For cases with small particles (B001, B003, B005), the particle radius parameter was set to 0.8 pixels, and for cases with big particles (B002, B004, B006), a value of 1.5 pixels was selected. Only particles with circular shapes were searched within the image. In other words, the shape parameters (A and B) were set to 1. The value of the cross correlation threshold for particle centroid location was set to 0.4, and the gray scale threshold value was set to zero. These values help ensure in the process of classifying pixels as part of a particle, the only discrimination will be the cross correlation threshold. Pixels touching each other and having a cross correlation value greater than the threshold will be considered as one particle. This improves the identification of particles that are too close together rather than using only the gray scale threshold. It was shown that for case B001 the number of vectors found by MaskPTV is greater than any other PTV program used in this work. This is due to the fact that MaskPTV is able to identify, with high accuracy, particles that are too close together. Figure 25 shows that the accuracy is almost inversely proportional to the concentration of particles. For low concentration, high accuracy and for high concentration, lower accuracy, the size of the particle showed almost no influence. The best case is B001, which corresponds to low concentration and large particles with almost 77% of vectors (11976 vectors) having an error less than 0.1.



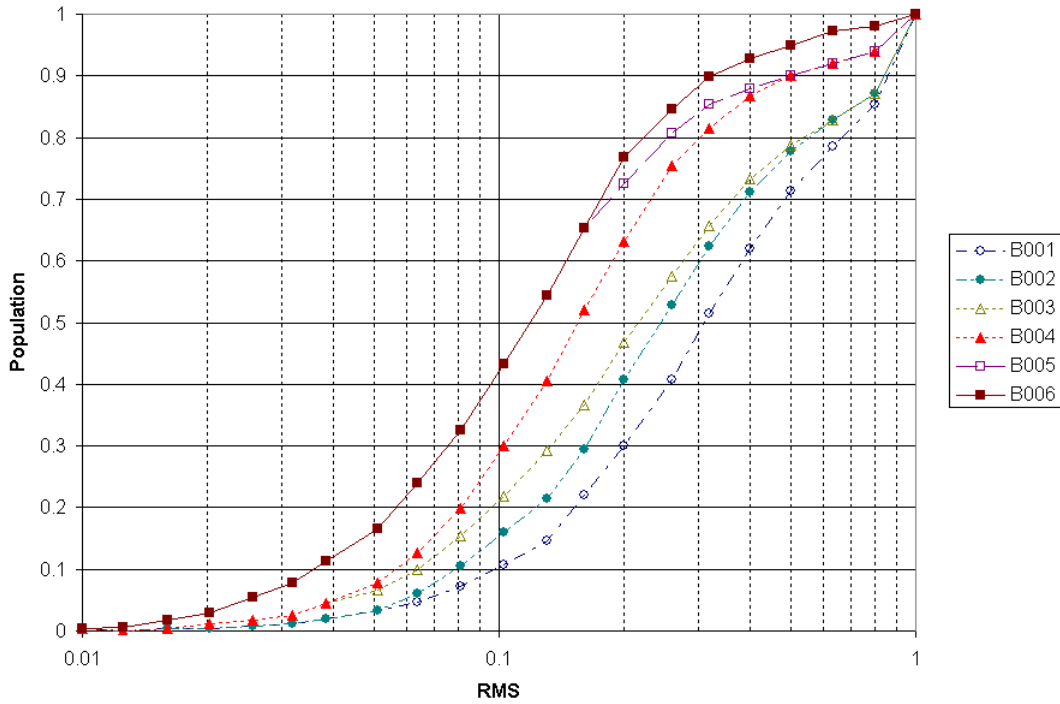
**Figure 25:** RMS probability density function of the displacement for different image concentration and sizes. Analysis with MaskPTV

### FspotsPTV Results

The same analysis by regions feature used in the MaskPTV program was used with the Fspots program, but in this case, a different centroid location algorithm based on edge detection and center of gravity is used. This procedure proved to be not as accurate as the ones used with MaskPTV. The use of the gray-scale threshold for the determination of the pixels that belong to a single particle seems to fail in the highly concentrated case, where only 2944 particles were detected with an RMS of 0.86. From the vectors detected, only 10% were below the range of 0.1 error.

Figure 26 shows that FspotsPTV behaves similarly to MaskPTV. This is expected because both programs share a similar particle tracking routine. For low particle concentration higher the accuracy is almost independent of the particle size.

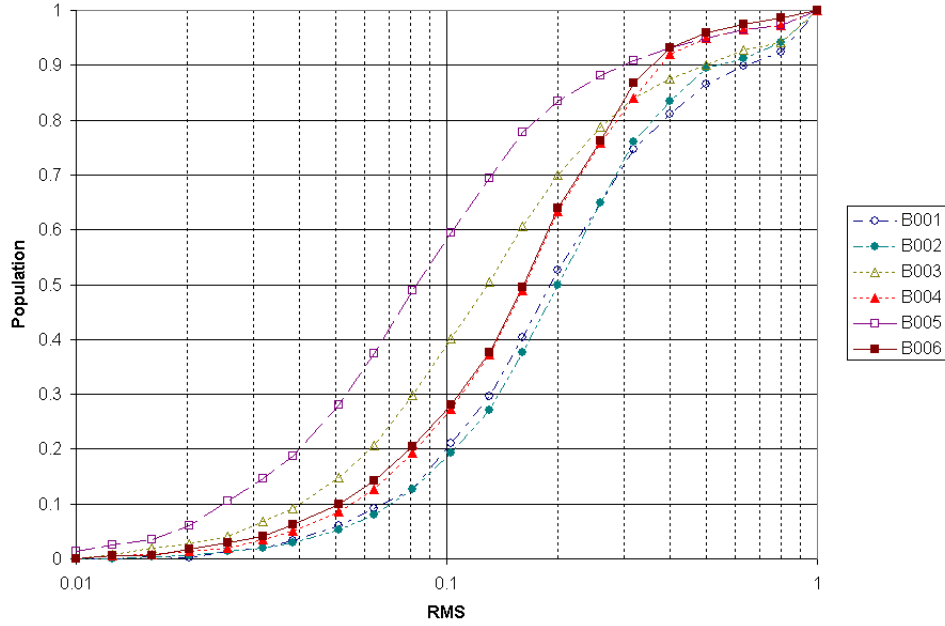




**Figure 26:** RMS probability density function of the displacement for different image concentration and sizes. Analysis with FspotsPTV

### AkaPTV Results

For this program, different regions of the same size as the ones generated on the previous PTV programs were used. Some of the parameters were modified in each region in order to obtain the largest quantity of vectors. The gray scale threshold was changed between the regions, depending not only on the quantity of vectors the program can handle, but also taking into account that if the number of particles is large, the array allocation will fail. This bug limits the performance of the algorithm. It was found to have a higher accuracy when using small particles (B005, B003, B001) (see figure 27). This is explained by the fact that this algorithm is strongly affected by the gray-scale threshold. Smaller particles tend to be more separated in the image than large particles, and this helps the algorithm to classify the pixels that belong to a particle.

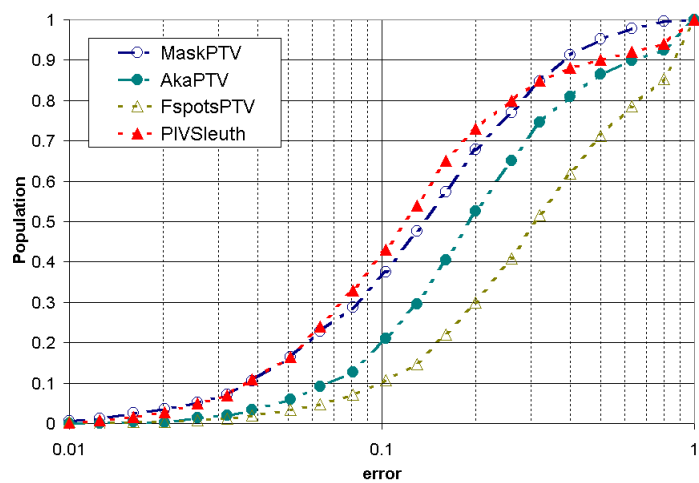


**Figure 27:** RMS probability density function of the displacement for different image concentration and sizes. Analysis with AkaPTV

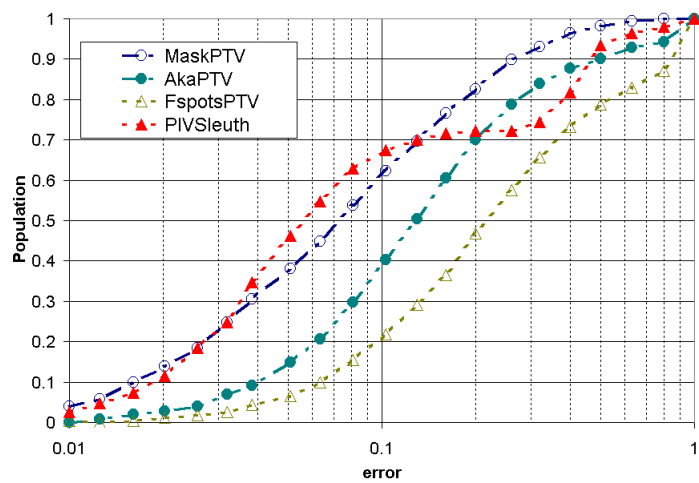
### Comparison Between Algorithms

In figure 28 and figure 29 the results and influence of the algorithms are summarized; MaskPTV together with PIVSleuth had the best results for high particle concentration images. However, for low concentration, PIVSleuth obtain a lower accuracy in contrast to MaskPTV. PTV algorithms are made to work with low concentration images; even with these “limitations”, MaskPTV (a PTV algorithm) also proved a trustable tool to use with highly concentrated images. AkaPTV program results are fairly good too. It found a large amount of vectors with high accuracy; nevertheless, it is the second most accurate PTV algorithm for small particles. For large particles, both algorithms AkaPTV and FspotsPTV behave almost the same, however, on the lower concentration images FspotsPTV had a better accuracy, but the drawback of locating a small number of vectors in all the cases remains prevalent.

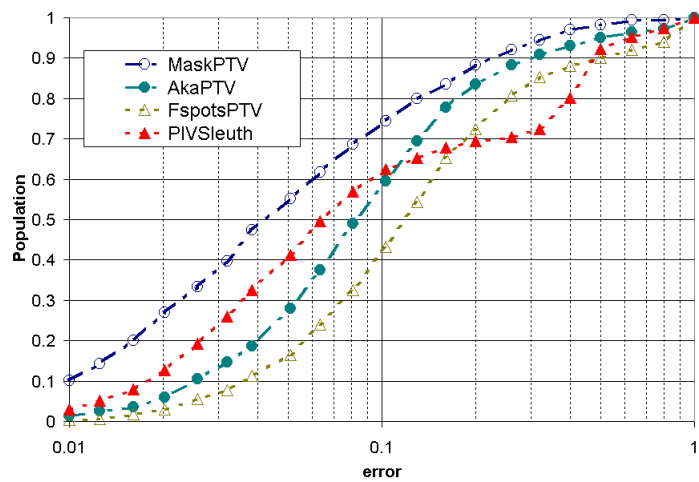
The velocity fields obtained by the different programs are shown in figure 30 and figure 31, case B001 (left pictures) and B006 (right pictures). These results show the difficulty that PTV algorithms face when dealing with high velocity gradients like the one found in the center of the vortex. The PIV algorithm seems to resolve the vortex center a little better.



(a)

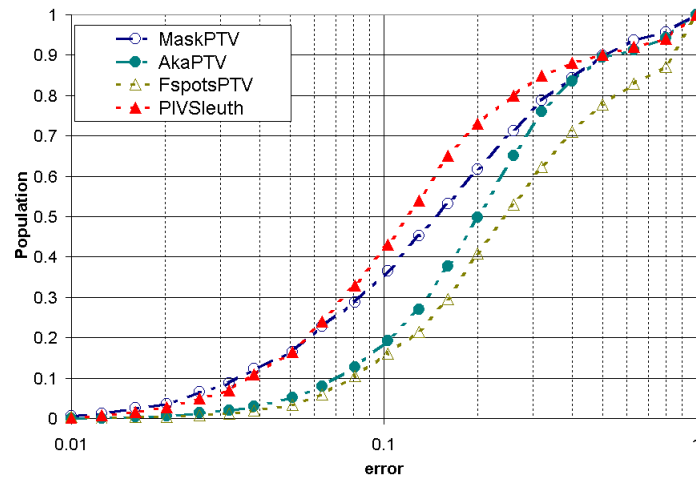


(b)

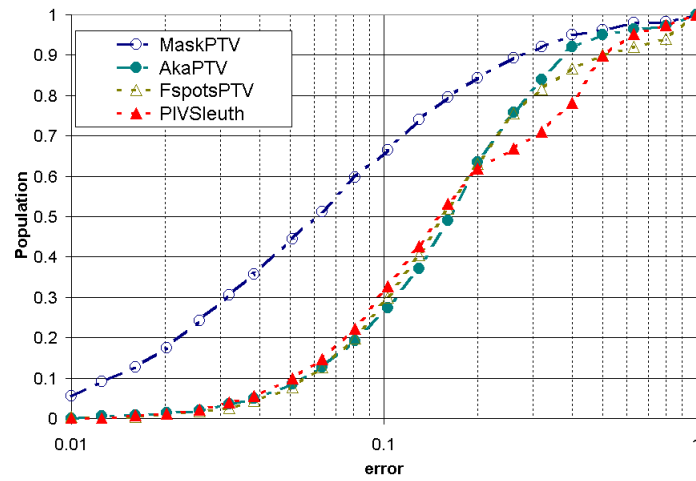


(c)

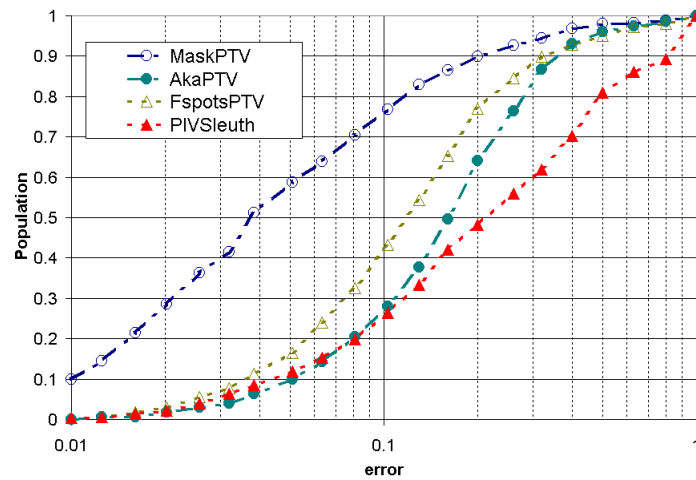
**Figure 28:** RMS probability density function comparison between the different image analysis techniques for case (a) B001, (b) B003, (c) B005



(a)

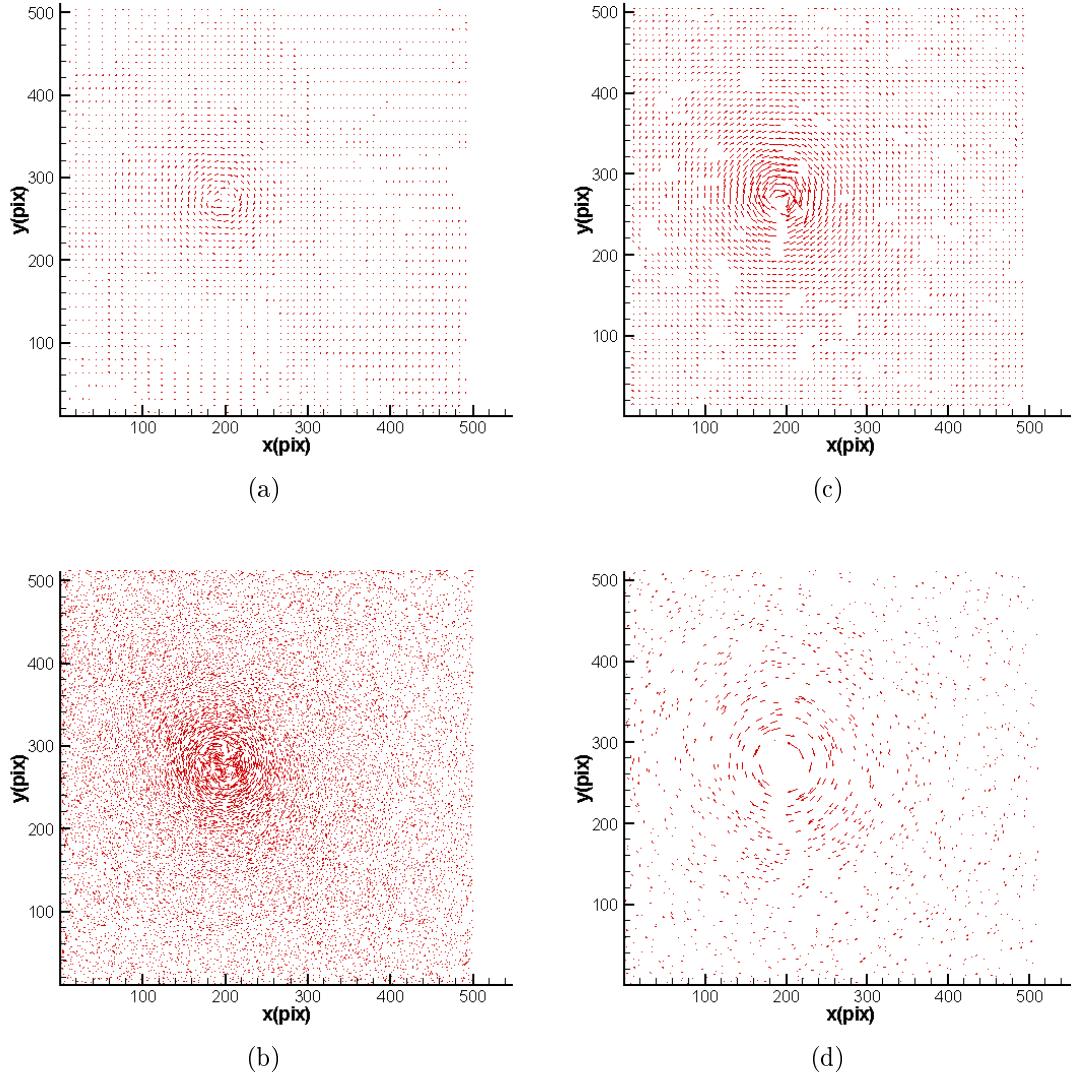


(b)

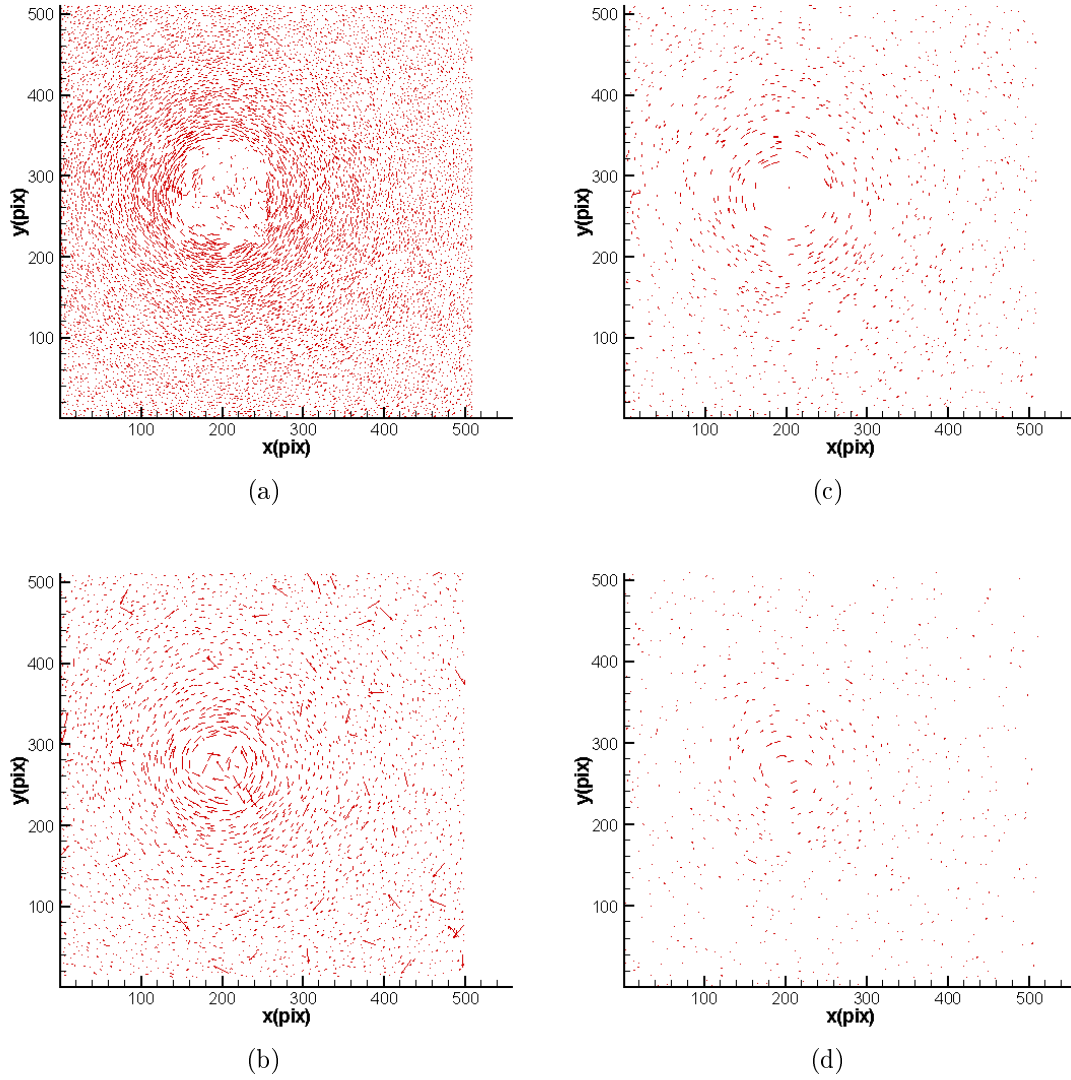


(c)

**Figure 29:** RMS probability density function comparison between the different image analysis techniques for case (a) B002, (b) B004, (c) B006



**Figure 30:** Velocity field obtained with (a) PIVSleuth, (b) MaskPTV for B001 case and (c) PIVSleuth, (d) MaskPTV for B006 case



**Figure 31:** Velocity field obtained with (a) AkaPTV and (b) FspotsPTV for B001 case and (c) AkaPTV and (d) FspotsPTV for B006 case

## EXPERIMENTAL RESULTS

The experimental images analyses results will be shown and discussed in a step by step order to give an idea of a “general procedure” to complete a successful and accurate PIV analysis. The first results are the sensitivity analysis used when changing programs parameters and is given in the next sections.

## Velocity Fields

Once the particles are detected correctly, the second step is to check if the velocity fields are what we are looking for. In the case of PTV, the beginning of the vectors must coincide with the centroid location of the object. Sometimes this test is easy to do by overlapping the resulting velocity field on the analyzed image.

Images from a two phase flow experiment were analyzed in order to compare the performance of the algorithms used in this study when trying to discern particles from bubbles. One important feature that makes PTV algorithms “better” than PIV algorithms is the characteristic of identification and tracking individual objects. Discerning bubbles from particles within the flow is one of the most important goals for any two phase flow research. Commonly researchers want to know parameters such as volumetric void fraction, void fraction distribution, relative velocity between phases, etc. All of these parameters can be measured using PTV; PIV is limited to obtaining only an average velocity, resulting in the lose of important detailed information contained on the travel history of each object within the flow. For this study, only MaskPTV and AkaPTV programs are tested because they have shown better accuracy in the analysis of artificial images.

**MaskPTV Results** As mentioned before, the mask cross correlation technique is implemented in the MaskPTV program to detect the centroid location of objects. It has the capability of searching for different object shapes within an image. In this section, this capability is used to classify objects in three categories: bubbles, particles, and background. Background identification is generally easy, assuming that pixels representing the background have a lower gray scale value than the pixels representing bubbles and particles. Applying a simple threshold on the gray scale level will differentiate the background from other objects. The classification of pixels that belong to bubbles instead of belonging to particles is a little more complicated; sometimes the average gray scale value of pixels representing bubbles is lower than the ones representing particles because of the particles reflective index of particles. Additionally the size of bubbles is generally larger than particles; the shape sometimes plays an important role as well, because bubbles will not always be as rounded as particles. Taking all this into account, an a priori analysis must be done in order to obtain a good search criteria.

**Object Location and Matching Between Frames** For the task of classifying objects, the search criteria of MaskPTV parameters are shown in table 1. In this case, all values are subject to the user experience; important information such as calibration (how many pixels per mm), average size of particles and bubbles, and image enhancing techniques used are needed. Image enhancing

algorithms tend to modify the size and gray scale values of objects.

**Table 1:** MaskPTV object classification parameters

Object	Radius	A	B	Mask1 Area	CCC1	MinPix	MaxPix	Min GS	Max GS
Particle	1.0	1.0	1.0	3	0.3	1	20	20	255
Bubbles	4.0	1.0	1.0	1	0.3	21	200	20	255
Background	—	—	—	—	—	—	—	0	19

For the task of matching objects between frames, the image was divided in three regions, each region having a different search area to locate matching candidates. Objects close to the wall have lower motion than the ones far from the wall, and the parameters of each region are shown in table 2. From these parameters, only the left search area is different between regions.

**Table 2:** MaskPTV object tracking parameters

Region	Search Left	Search Right	Search Up	Search Down	CCC2	Xi	Xf	Yi	Yf
1	6	2	2	2	0.4	0	1008	0	25
2	9	2	2	2	0.4	0	1008	25	50
3	25	2	2	2	0.4	0	1008	50	1008

Using the previous parameters for the MaskPTV program, the resulting velocity field for the two phases is shown in figure 32(a), where the red arrows represent the velocity of the particles, and blue arrows represent the velocity of bubbles. The bubble location is represented with the blue circles. 3384 particles and 155 bubbles were found using the previous parameters.

**AkaPTV Results** The parameters used for object classification are summarized in table 3. The AkaPTV program uses the gray scale level threshold and object size to classify pixels into background, particles, and bubbles. The difference between objects is mainly based on the object size and is strongly influenced by the user criteria.

**Table 3:** AkaPTV object classification parameters

Object	MinPix	MaxPix	MinGS
Particle	1	20	150
Bubbles	21	200	150
Background	—	—	< 150

Parameters for the tracking routine are summarized in table 4. In this case, 3 regions with

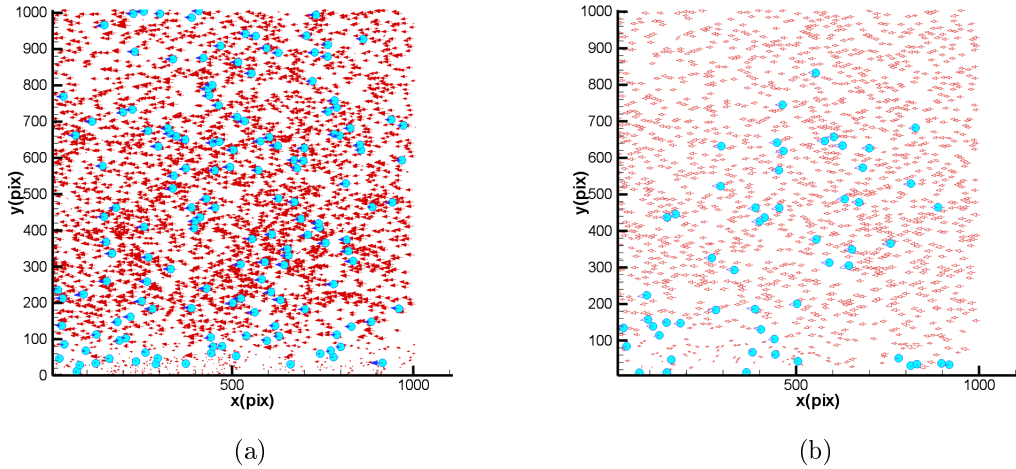


different searching areas were selected to reduce the number of possible candidates in order to make a matching pair of objects between frames.

**Table 4:** AkaPTV object tracking parameters

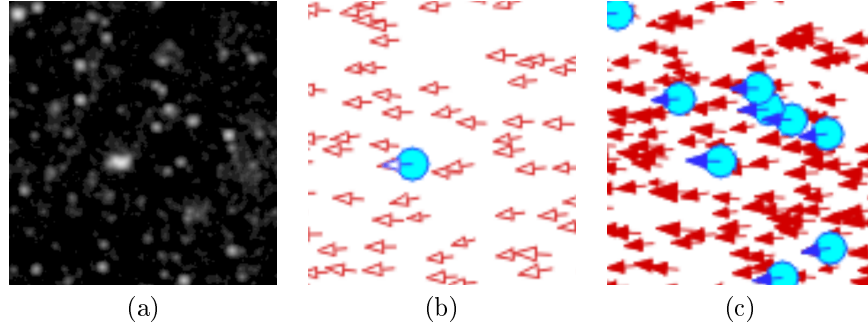
Region	Search Area	Cluster Size	Grid	Xi	Xf	Yi	Yf
1	6	15	28	0	1008	0	25
2	9	15	28	0	1008	26	50
3	25	15	28	0	1008	50	1008

Using the previous parameters for the AkaPTV program, the resulting velocity field for the two phases is shown in figure 32(b), where white arrows with red border represents the velocity of the particles, and blue arrows represents the velocity of bubbles. The bubble location is represented with blue circles. 55 bubbles and 1509 particles were found using the previous parameters.



**Figure 32:** Velocity field of the two phases and bubble location using (a) MaskPTV algorithm, and (b) AkaPTV algorithm

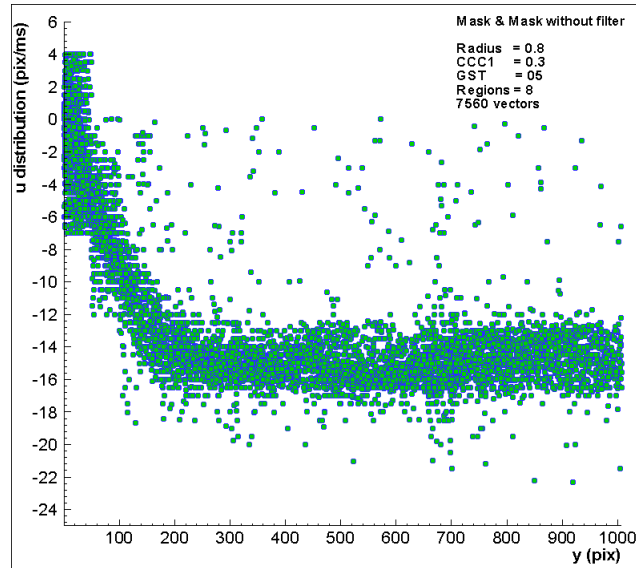
The sensitivity of the previous parameters is better explained in figure 33, where the difference between the resulting velocity fields are shown in detail. Figure 33(b) shows the resulting velocity field from AkaPTV, and figure 33(c) shows the results from MaskPTV. The MaskPTV program shows more bubbles, but this is only because it used a less restrictive search criteria than the used in the AkaPTV program. The difference between the number of particle vectors found is because MaskPTV has the characteristics to discern accurately two particles closely together, thus a the greater amount of vectors.



**Figure 33:** Comparison of the resulting velocity fields between (b) AkaPTV program and (c) MaskPTV program obtained with the analysis of (a) original image

### Analysis of a Single Field

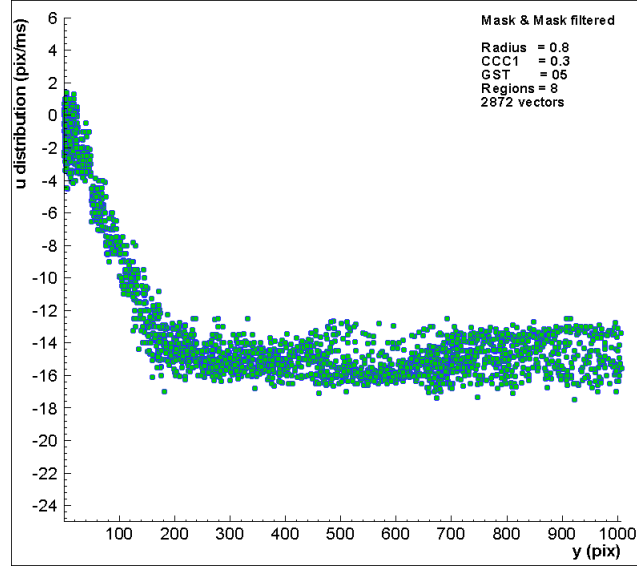
Once a single velocity field is obtained with the parameters selected, we would like to know if these parameters are correct before analyzing the whole set of images. This can be done by checking the information that we already have. For the experiment described in the previous sections, the most useful distribution will be the  $u$  instantaneous velocity vs the  $y$  coordinate, and this is shown in figure 34.



**Figure 34:** Instantaneous  $u$  velocity distribution vs  $y$  coordinate

The  $u$  instantaneous velocity distribution in figure 34 shows the three regions in which the image was divided in order to reduce the search area to track objects. It's clear that the lowest  $u$  velocities are found close to the wall. Spurious vectors were found, but they can be easily eliminated using a

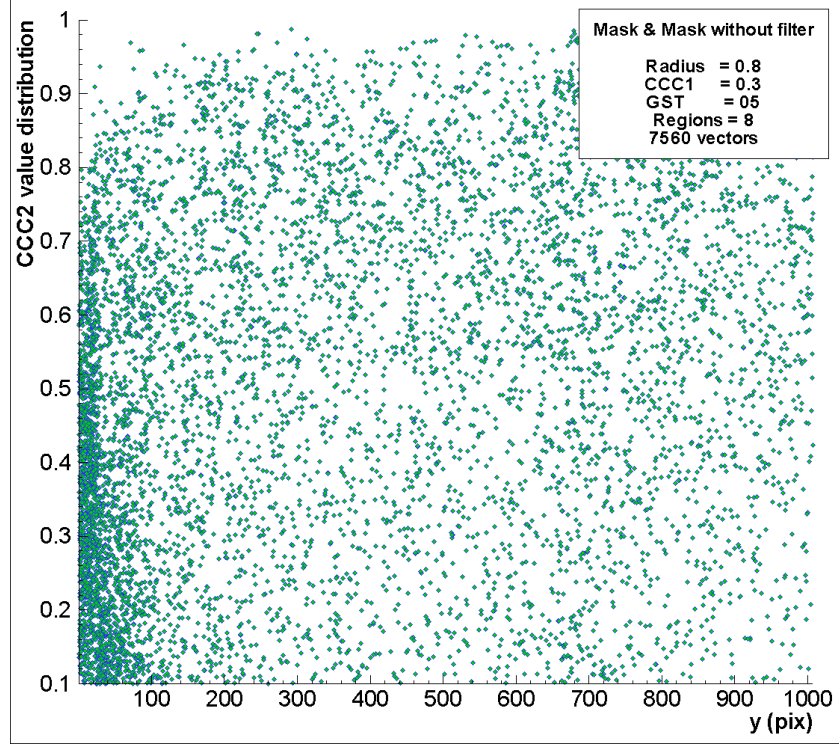
filter algorithm. In figure 35, the results of applying a statistical filter are shown.



**Figure 35:** Filtered  $u$  instantaneous velocity distribution vs  $y$  coordinate

From the filtering procedure a considerable amount of vectors are eliminated to ensure a final result without spurious vectors. As can be seen by comparing figures 34 and 35, “good” vectors were also erased. The quantity of images per set will determine the restrictiveness of the filtering algorithm. If enough velocity fields are available and the velocity distributions seem fairly good, the filtering process can be omitted.

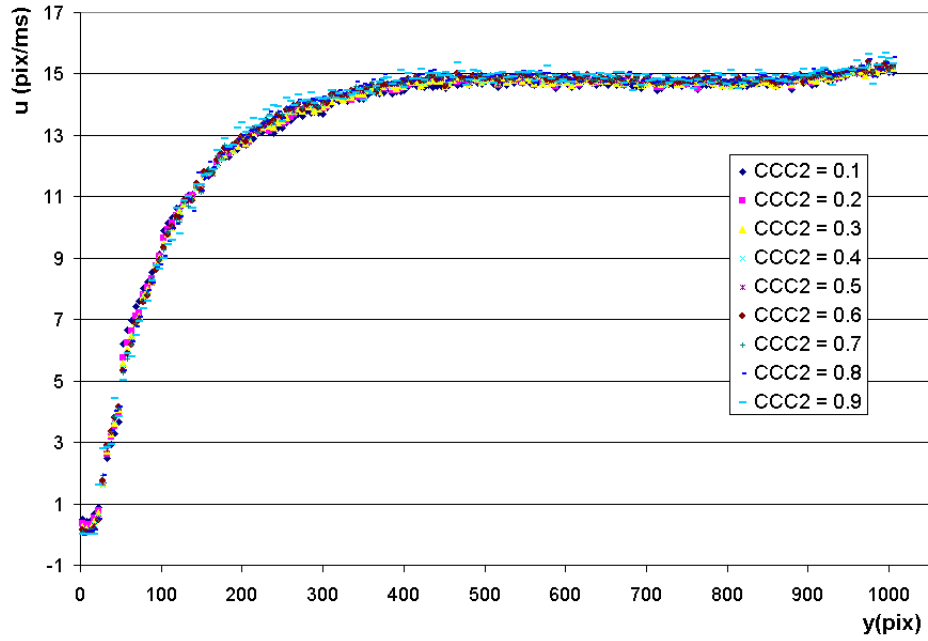
Distributions of other quantities can also be used. For example, the  $v$  velocity distribution vs  $x$  coordinate can be plotted also, but in this particular case the information obtained was not useful. There is not a preferred  $v$  velocity along the  $x$  coordinate. The  $C_2$  distribution can also be plotted to give an idea in which region of the image the highest errors are found. In figure 36(a), the distribution of  $C_2$  vs  $y$  coordinate is plotted, and it shows that the highest errors are found close to the wall. The area close to the wall will experience greater reflections and intensity of the beam, and this indicates that we must be more restrictive on the parameters used on regions close to the wall to help reduce the amount of spurious vectors.



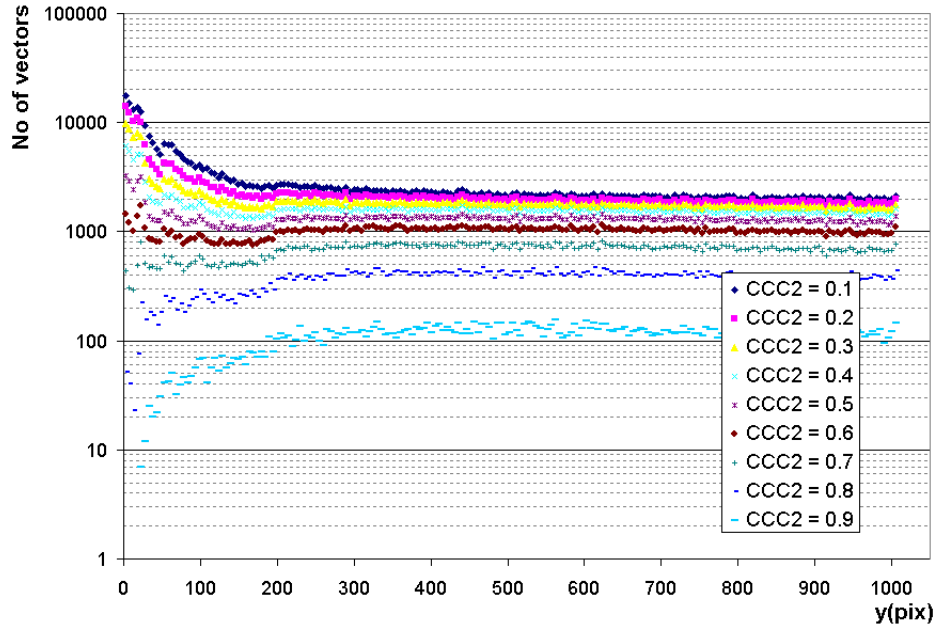
**Figure 36:**  $C_2$  distribution over the  $y$  axis

### Statistical Analysis

After a single field is analyzed, the next step is the use of statistical quantities to validate our results. In order to have good statistics, a larger number of data is required; therefore, the whole set of images from a given experiment must be used and from these results we can extract average quantities. For example, in figure 37(a), the influence of different values of the  $C_2$  threshold is tested. As can be seen, there is not a significant change on the velocity profile. This is a good sign that the parameters chosen for the analysis are correct. In figure 37(b) it can be seen that the  $C_2$  threshold modifies greatly the number of vectors found in the analysis. Assuming that the flow is uniformly seeded, this plot should be flat. A good value for this flatness could be 0.65, but in doing so, the global statistics of the analysis will be compromised due to the decreasing number of vectors.



(a)

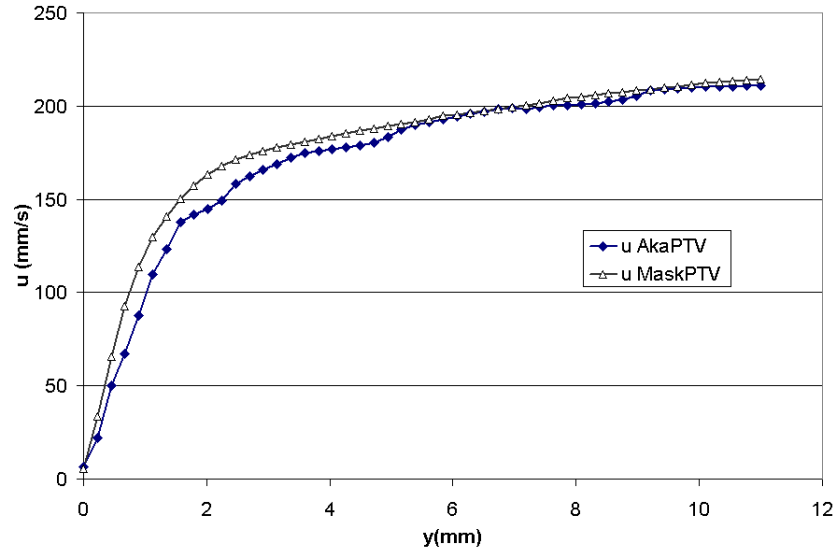


(b)

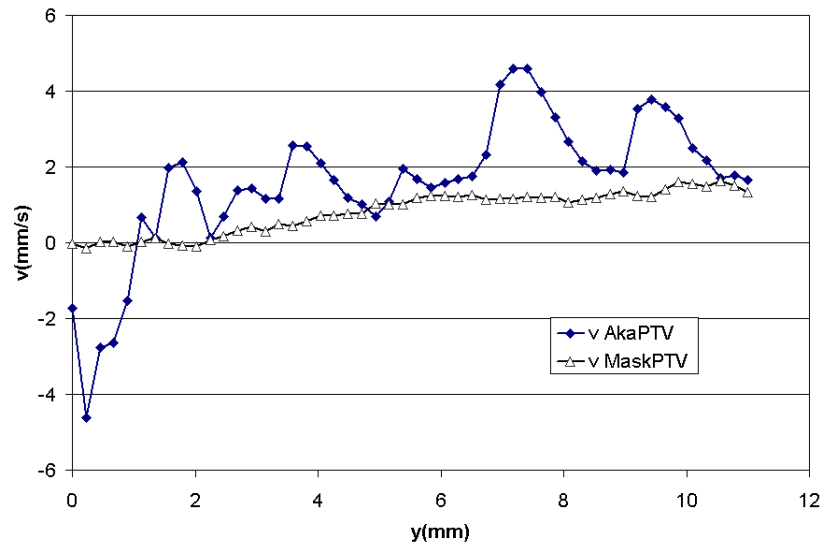
**Figure 37:** (a) Influence of the  $C_2$  over the average in time of  $u$  velocity profile, (b) influence of different  $C_2$  over the number of vectors found on the whole set

The parameter sensitivity analysis explained in the previous paragraphs does not ensure that we will obtain a correct result, and therefore, the rest of the statistical analysis must be done. In

figure 38 the velocity profiles for the  $u$  and  $v$  velocities are shown, and they seem to be correct. The little bumps on the results of AkaPTV program must give us a warning that something is not right on the parameters chosen.



(a)

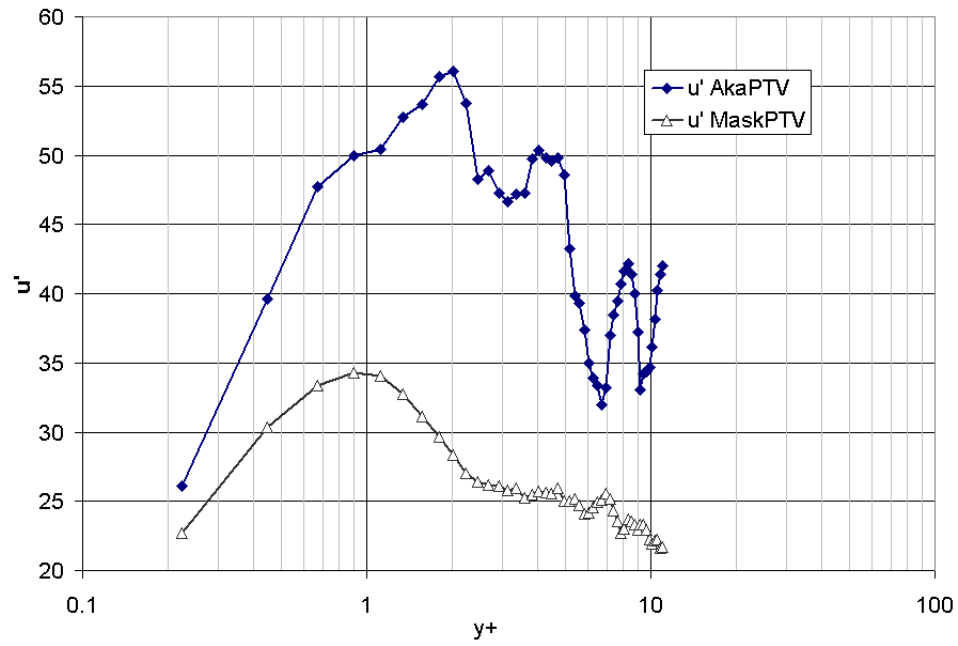


(b)

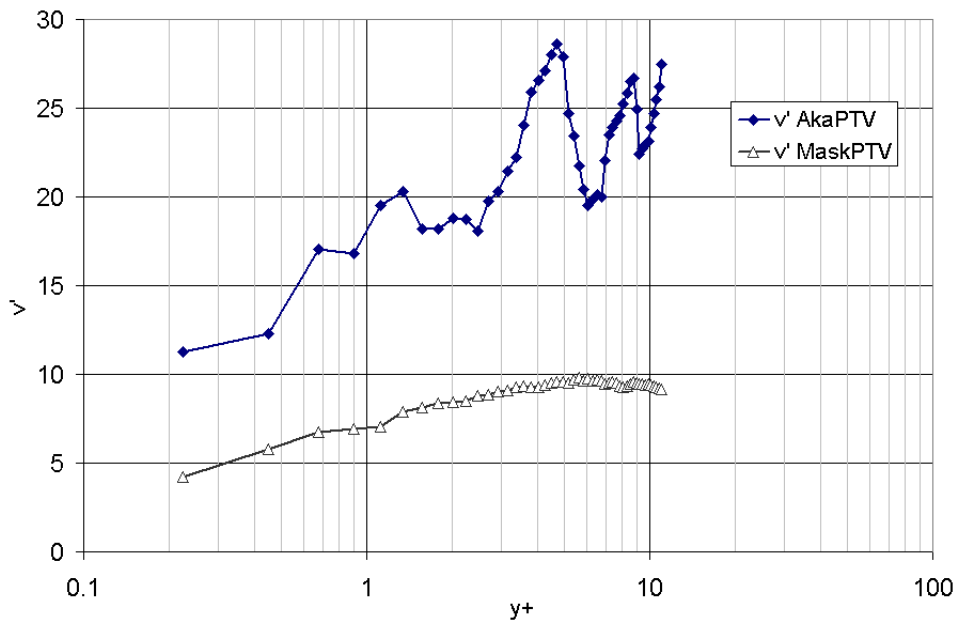
**Figure 38:** (a) Velocity profile  $u$  obtained from MaskPTV and AkaPTV programs, (b) velocity profile  $v$  obtained from MaskPTV and AkaPTV programs

This error is more evident when calculating the  $u'$  and  $v'$  turbulence intensities shown in fig-

ure 39, and the continued propagation in calculation of the Reynolds Stresses  $u'v'$  shown in figure 40.

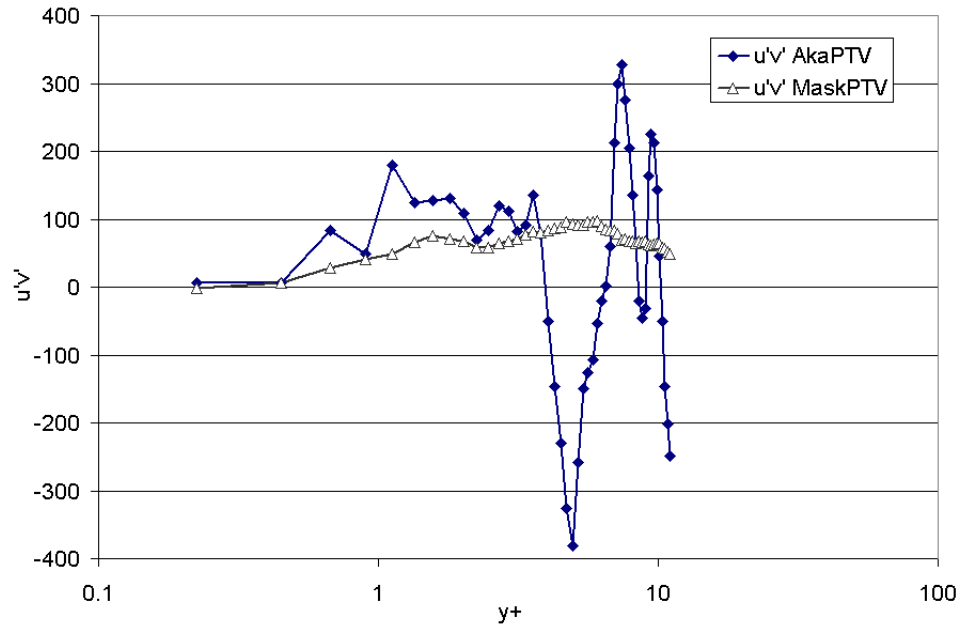


(a)



(b)

**Figure 39:** (a)  $u'$  turbulence intensity from MaskPTV and AkaPTV programs, (b)  $v'$  turbulence intensity obtained from MaskPTV and AkaPTV programs



**Figure 40:**  $u'v'$  Reynolds stresses obtained from MaskPTV and AkaPTV programs



## CONCLUSIONS

In this work, a Windows program based on particle tracking algorithms was developed, and its accuracy was systematically investigated for a large variety of image and flow parameters. Since the practical use of PIV requires the processing of a large number of images, a substantial effort was devoted to develop robust algorithms which provide reliable results over a wide range of inputs. Furthermore, these algorithms were improved thanks to the knowledge gained in the laboratory experience, which took into account the reduction of human time consuming processes which are solved automatically by the program. The coupling of the algorithms into a single system makes the use and input of information even easier, providing immediate visual results to help decide whether or not the parameters entered are correct. Even if the computing time employed by the new program is much larger for example than the high speed algorithm of AkaPTV, there are time savings because once the program is started it will not need any human supervision. In contrast, the AkaPTV program version is buggy and fails constantly. The capability of analysis by different regions makes MaskPTV program even better than AkaPTV, because AkaPTV has to deal with each analysis of region one at a time, increasing the human time expenses.

On the comparison of several PIV algorithms, the use of PIV challenge standard images ensures that there is not a preferable bias in the creation of the test images to benefit a given algorithm. In general, all algorithms perform fairly well, with the exception of the Fspots centroid location algorithm that gave the lowest accuracy and quantity of vectors from all the programs. MaskPTV proved to work as good as PIVSleuth even with images generated specifically for PIV algorithms, i.e. highly concentrated flows with large particle sizes. No other algorithm was capable of giving as high a number of vectors with the accuracy of the MaskPTV program. It can be concluded that the algorithms developed in this work increased the accuracy of the former Fspots and tracking algorithms with an easy to use windows interface. Unlike other studies where the accuracy of the results is inferred indirectly by some error analysis, the accuracy of the processing was evaluated directly in this study by direct comparison with artificial values. In this fashion, the errors were calculated instead of inferred. Statistics of the errors were compiled by processing large data sets for good statistical convergence. The sensitivity of the results to the various image parameters were then systematically evaluated. These extensive tests also validated the robustness of each algorithm. The parameters included the number of particles in the image, their average size, and the noise level in the image. Since the accuracy with which the velocity can be determined is intrinsically related to the accuracy with which particles can be located, most of the analyses were performed on the errors involved in locating the centroid of the objects.

The dominating factor controlling the validity of the processing is the ability to separate the particles from the background. To that effect, the most important part of the new algorithm was to choose not one, but two threshold values, the gray scale and the centroid cross correlation thresholds. The failure of the selection of these thresholds results in the detection of a very large number of spurious objects. Therefore, experience and common sense helps with the determination of these values for a given problem.

It was also found that the error increases as the mean radius of the particle increases (for a fixed number of particles). Once again, this relates to the higher likelihood of particle overlap. In practical PIV, particle overlap is controlled by the width of the light sheet illuminating the particles, and it can be practically eliminated with sharply collimated laser light sheets. The concentration has the same effect. The error is proportional to the object concentration but with lower consequences. Another parameter that greatly influences the performance of the algorithms is the image noise, which drastically changes the results from a clean picture. It does not affect only the spurious location of particles, but also affects the correlation value used to match particles between frames. Therefore, when using real images, an image processing must be selected in order to reduce the noise.

The extension of the error estimation using artificial images to real experimental images can be done by generating images that mimic the experimental ones to such an extent that even experimental noise and reflections can be reproduced into the artificial images. Following the procedures of the analyses stated in this work, a good error estimation can be obtained even for the two phase flow experiments. The functional relationship between the correlation coefficients and the error calculated can be used to estimate experimental errors.

The follow up analysis for experimental images developed in this work can help to reduce the time expensive trial and error procedure of selecting good parameters by first analyzing a single field to verify the input parameters before the whole set of images is analyzed.

Future work is still needed to optimize the algorithm with an accurate high speed program capable of the same order of accuracy, and a possible extension to 3D PTV is also feasible. Also, the extension to high resolution PTV by means of an educated searching area based on the results of a previous PIV analysis can be developed.

## REFERENCES

- Adrian R J 1984 Scattering Particle Characteristics and Their Effect on Pulsed Laser Measurements of Fluid Flow: Speckle Velocimetry vs. Particle Image Velocimetry *Applied Optics* **23**(11) 1690-1691
- Adrian R J 1986 Image Shifting Technique to Resolve Directional Ambiguity in Double Pulsed Velocimetry *Applied Optics* **25**(21) 3855-3858
- Bradshaw P 1971 *An Introduction to Turbulence and Its Measurement* (New York: Pergamon Press LTD)
- Brodkey R S 1967 *The Phenomena of Fluid Motions 3rd ed.* (New York: Addison Wesley)
- Brown G and Roshko A 1974 On Density Effects and Large Structure in Turbulent Mixing Layers *J. Fluid Mech.* **64** 775-816
- Corino E and Brodkey R 1969 A Visual Investigation of the Wall Region in Turbulent Flows *J. Fluid Mech.* **37** 1-30
- Dimotakis P, Debussy F and Koochesfahani M 1981 Particle Streak Velocity Field Measurements in a Three-Dimensional Mixing Layer *Physics of Fluids*. **24** 995
- Frost W and Mouldin T 1977, *Handbook of Turbulence Volume 1*, (New York: Plenum Press)
- Goldstein R 1983 *Fluid Mechanics Measurements* (Berlin: Hemisphere Publishing Corporation)
- Hassan Y, Blanchat T and Seeley C 1992 PIV Flow Visualization Using Particle Tracking Techniques, *Meas. Sci. Technol.* **3** 633-642
- Head F and Bandyopadhyay P 1981 New Aspects of Turbulent Boundary Layer Structure *J. Fluid Mech.* **107** 297-337
- Hinze J 1975 *Turbulence 2nd ed.* (New York: McGraw-Hill)
- Kobayashi T and Ishihara Y 1985 An Automated Analysis Method for Determining Velocity Vectors from Path-line Photograph *Proc. Int. Symposium on Fluid Control Meas.* 729-734 (*Tokyo, Japan*)
- Lourenco L 1986 Theory and Applications of Particle Image Displacement Velocimetry *Lecture Series, von Karman Institute for Fluid Dynamics, (Belgium)*
- Mossi M and Sagaut P 2003, Numerical Investigation of Fully Developed Channel Flow Using Shock-Capturing Schemes *Computers & Fluids* **32** 249-274

- Nishino K and Kasagi N 1985 Flow Visualization and Its Digital Image Processing in a Fully Developed Two Dimensional Turbulent Channel Flow *Proceedings of the Int Symp. on Fluid Control Meas. (Tokyo, Japan)*.
- Nychas S, Hershey H and Brodkey R 1973 A Visual Study of Turbulent Shear Flow *J. Fluid Mech.* **61** 223-239
- Offen G and Kline S 1974 Combined Dye-Streak and Hydrogen-Bubble Visual Observations of a Turbulent Boundary Layer *J. Fluid Mech.* **62** 223-239
- Okamoto K, Nishio S, Saga T and Kobayashi T 2000 Standard Images for Particle Image Velocimetry *Meas. Sci. Technol.* **11** 685-681
- Okamoto K, Stanislas S and Kahler C 2003 Main Results of the First International PIV Challenge *Meas. Sci. Technol.* **14** R63-R68
- Perry A 1982 *Hot Wire Anemometry* (Oxford: Clarendon Press)
- Praturi A and Brodkey R 1978 A Stereoscopic Visual Study of Coherent Structures in Turbulent Shear Flow
- Riethmuller M 2001 *Particle Image Velocimetry and Associated Techniques* (Belgium: von Karman Institute)
- Russ K 1988 Particle Color Considerations in Flow System Image Acquisition *M. S. Thesis. The Ohio State University*
- Tennekes H and Lumley J 1972 *A First Course in Turbulence* (Cambridge, MA: MIT Press)
- Tetsuo S, Shigeki S, Tshio K and Hui H 2001 Development and Evaluation of an Improved Correlation Based PTV Method *Journal of Visualization* **4**(1) 29-38
- Tsubokura M, Kobayashi T and Taniguchi N 1997 Visualization of 3-D Structures in a Plane Impinging Jet Using Large Eddy Simulation *Proc. FLUCOME'97* vol 2, p 875
- Uemura T and Yamamoto F 1993 A High Speed Algorithm for 2D and 3D and Its Application. *The 6th International Symposium on Transport Phenomena in Thermal Engineering. (Seoul, Korea)*
- Van der Plas, G Bastians R 1998 Accuracy and Resolution of a Fast PTV-Algorithm Suitable for HiRes-PV *Proceedings 8th International Symposium on Flow Visualisation (Sorento, Italy)*
- Yeh Y and Cummins H 1964 Localized Fluid Flow Measurements with an He-Ne Laser Spectrometer *Applied Physics Letters* **4** 176-178

## APPENDIX A

### Example of Euler's Approximation to Obtain the Directional Field

To generate a directional field of  $11 \times 11$  points for  $y' = x + y(1 - y)$  on  $-1 \leq x \leq 1, -2 \leq y \leq 2$ , the first step is to let  $f(x, y) = x + y(1 - y)$ , the 121 grid points are the pairs  $(x, y)$ , where the minimum distance between grid points is  $H = 0.2$ .

Generating the endpoints of the replacement line segment at  $x_i = -0.4, y_i = 1.6$ . It will be shown that the first endpoint is  $(-0.341, 1.519)$ . This point can be located from  $(x_i, y_i)$  by travelling a distance  $H/2$  at slope  $M = -1.36$ .

$M = f(x_i, y_i)$  The line segment slope for Euler's rule.

$M = x_i + y_i(1 - y_i)$  Apply  $f(x, y) = x + y(1 - y)$

$M = -1.36$  Use the first point  $x_i = -0.4, y_i = 1.6$

$h = H/(2\sqrt{1 + M^2})$  Apply the formula  $h = H/(2\sqrt{1 + M^2})$

$h = 0.0592$ , Use  $H = 0.2$  and  $f(x_i, y_i) = M = -1.36$

$x_f = x_i + h$  Compute the  $x$ -coordinate of the second point.

$x_f = -0.341$  Use  $x_i = -0.4$  and  $h = 0.0592$ .

$y_f = y_i + hf(x_i, y_i)$  Compute the  $y$ -coordinate of the second point.

$y_f = 1.519$  Use  $y_i = 1.6, f(x_i, y_i) = M = -1.36$  and  $h = 0.0592$ .

Automation of this process is necessary because 121 such calculations are required.

## **VITA**

### **Name**

Carlos Eduardo Estrada Pérez

### **Address**

Rubén Darío No.77,

Col. Moderna, Del. Benito Juárez,

México, D.F. C.P. 03015

### **Education**

Bachelor of Science in energy engineering

obtained in Mexico City

at the Universidad Autónoma Metropolitana, 2000