Design of a software architecture for Velocimetry Systems

ROGER PIZZATO NUNES[1]
FLÁVIO TADEU VAN DER LAAN[2]
[1]Departamento de Física – Instituto de Física – Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves 9500, Caixa Postal 15051, CEP: 91501-970 – Porto Alegre, RS, Brasil.
roger.pizzato@ufrgs.br or rogerpn@if.ufrgs.br

[2]Depto. de Eng. Nuclear – Escola de Engenharia – Universidade Federal do Rio Grande do Sul
Av. Osvaldo Aranha 99, 4o. Andar, CEP: 90046-900 – Porto Alegre, RS, Brasil.
ftvdl@ufrgs.br

Abstract. The physical systems in the area of fluid mechanics present a complex behavior and, in certain regimes, they have not been completely described and understood by the physics laws yet. For this reason, the experimental techniques employed for the velocity field measurement of a given fluid flow still play an important hole. It is critical in Engineering, a knowledge area in which it is necessary to project, to implement, or to validate a prototype that involves such physical concepts. In the last ten years, the technological advance in the areas of microelectronics, optics and computer science allowed that image processing techniques were applied to the analysis of fluid flows. These techniques have propitiated that much more qualitative and quantitative information could have been extracted from the fluid flow. As a consequence, a better and a more accurate characterization of the physical phenomena became possible, enabling the validation of the developed models. In this context, the main objective of this work is to show the architecture of a computational tool for the measurement of the two-dimensional velocity field of fluid flows through digital image processing. This tool has been applied in different physical systems in fluid mechanics, providing great results as our recent publications show.

Keywords: image processing, flow visualization, and velocimetry systems.

(Received October 24, 2006 / Accepted June 05, 2007)

1 Introduction

In the macroscopic view, the mathematical description of the state of a moving fluid is effected by means of functions which give the distribution of the fluid velocity $\mathbf{v}(\mathbf{r},t)$ and of any two thermodynamic quantities pertaining to the fluid, like the pressure and density, forming a hydrodynamic and thermal coupled problem. If these five quantities are determined, namely the 3 components of the velocity field $\mathbf{v}$, the pressure, and the temperature, the dynamics of the fluid is completely described. For each one of these quantities, there is a nonlinear partial differential equation that has to be solved for a specific boundary condition [1].

The paragraph above was intended to contextualize the complexity of the physical systems in the fluid mechanics. Generally, nonlinear boundary value problems involving more than one spatial dimension do not have analytical solutions. Fortunately, in most cases, it is not necessary to obtain the entire information about the fluid flow that analytical solutions provide, but just extracting from them some specific information that is relevant in the current investigation. This is the hole of the models, which intend to describe the physical mechanism under research to achieve a physical insight and interpretation about the problem.

However, these models must be validated and here the importance of experimental techniques emerges. To verify if the predictions of the developed model are consistent with the real physical system, it is necessary to compare its results with experimental data. How much more the results of the models approximate of the experimental results, more accurate are the models to describe the physical phenomena. Nowadays, in the academic scope, these experimental results have been much used for the validation of models that describe the mechanisms involved in the turbulence [8, 9], a problem that has not been understood theoretically by the classical physics yet [3]. In industry, these results are employed to achieve necessary empirical mathematical relations in projects of engineering in the areas of aerodynamics [4], turbomachinery [5], biomedical [6], plasma technologies [2], metallurgy [7], and others.

Considering an incompressible flow and the regimes where the thermodynamic effects can be neglected, the instantaneous velocity field of a fluid flow is the more important and fundamental physical quantity in the fluid dynamics. With the instantaneous velocity field, it is possible to compute flow quantities like vorticity, intensity of turbulence, circulation, and others. These quantities completely determine the characterization of the fluid flow under analysis [13].

In the last years, many experimental techniques had been developed to measure the velocity of a fluid flow [10]. However, the evolution of the image sensors like the CCD (Charged Coupled Device) allowed that velocimetry techniques through digital image processing were developed [11]. The main profit of these techniques in relation to the traditional ones is the possibility for a given time $t=t_o$, to measure a whole velocity field $\mathbf{v}(\mathbf{r}, t_0)$ in the region of $0 \le \mathbf{r} \le \mathbf{r}_o$, instead of just one velocity vector in a specific position $\mathbf{r} = \mathbf{r}_o$. This is due to the two-dimensional characteristic of the electronic image sensor, which allows that a certain area (instead of just one point) of the fluid flow can be analyzed for a particular instant of time [12].

The purpose of the present paper is to show the design of a software architecture for measurement of the two-dimensional velocity field of arbitrary fluid flows. This software, whose name is IPVFlow, version V1.0, is based on digital image processing algorithms and is part of an instrumentation system for velocimetry of fluid flows [12]. This tool was successfully applied by us in many mechanical systems in fluid dynamics, as fluid flows inside circular pipes [16] and fluid flows inside metallurgical ladles [7, 19]. Since some of these algorithms have been already published by us [17], the goal of this work is not to present a detailed description about the image processing algorithms, but showing how these algorithms interacts with each other in a global point of view. This approach is essential and helps to understand how the computational tool was developed to satisfy its necessary requirements and how it can be improved to achieve better results.

The paper is organized in the following form: in section 2, the general principle of velocimetry and a brief explanation about the overall instrumentation system used to capture the fluid flow frames are presented. In section 3, the main algorithm that governs all the IPVFlow V1.0 software is described. In section 4, the results of the IPVFlow V1.0 software are shown. For this, some frames from 3 different fluid flows were selected and processed by the IPVFlow V1.0, being its measured velocity fields exposed. In section 5, the conclusion of this paper and the perspectives for future works will be discussed.

2 General principle of velocimetry

The principle of a velocimetry system through image processing consists in determining the distance covered by one and/or by an ensemble of particles on a well defined interval of time. With this objective, some micrometric particles are spread in the fluid flow. These tracer particles are assumed enough small and with the same density of the fluid, so that, respectively, the drag force and the buoyancy force can be neglected. Also, these particles perfectly follow the streamlines of the fluid flow, implying that the lift forces are ignorable. Synthetically, it can be supposed that the particles do not interfere in the fluid flow, acting simply as a tracer [12].

To capture the information from the fluid flow, the tracer particles are illuminated by a coherent light source (usually a laser) and its images are captured by a camera with an electronic image sensor (usually a CCD but most frequently now a CMOS). The camera captures successive frames of the particle ensemble trajectory with a stable time-base Δt. The frequency with which the frames are captured from the fluid flow is known as frame-rate, represented here by T_q.

An example of a typical instrumentation system employed for the measurement of velocity fields of fluid flows is shown in Figure 1. It is composed by a laser source, optical lens, tracer particles, electronics circuits for triggering the equipments, a CCD camera and a Personal Computer (PC). The trigger device was used to synchronize the laser source with the CCD camera for frame acquisition. Also, this equipment has a relevant function in periodic fluid flows or in fluid flows initiated by an external event [12].

It is important to notice that each frame represents the spatial behavior of the fluid flow in a particular instant of time, while the sequence of frames provides the information about the fluid dynamics. Then, the spatial structures of the fluid flows can be calculated and evaluated as time evolves.

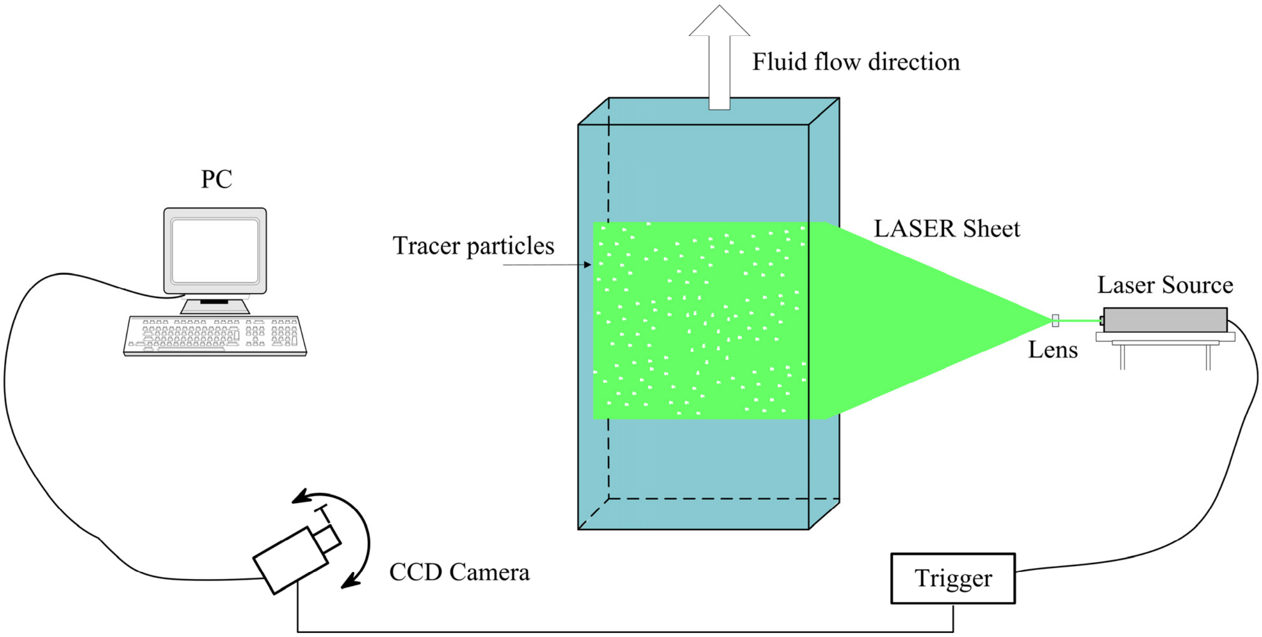


Figure 1 – The instrumentation system employed in velocimetry of fluid flows.

In this sense, the image processing techniques provides a three dimensional measurement, two of them are spatial and one is temporal. For the same optical devices, the spatial and temporal resolutions are directly related with the electronic image sensor characteristics. In a simple way, how much larger the pixel density in the CCD sensor, more spatial resolution will have the measurement. In the same way, how much higher the frame-rate T_q, more temporal resolution the measurement will have.

Analyzing the experimental apparatus of the Figure 1, which is applied over an arbitrary physical system (the fluid flow), it can be seen that a velocimetry system can be segmented in two functionally distinct subsystems, named here of the image formation subsystem (IFS) and the image processing subsystem (IPS). The task of IFS is to form and to capture the images from the fluid flow. This subsystem is composed by the tracer particles, the laser source, trigger circuits for synchronization, optical lens, and a camera with an electronic image sensor. In resume, the subsystem IFS just involves the equipments used in the instrumentation. On the other hand, the IPS subsystem just comprises the PC, where the frames captured from the fluid flow are stored for future analysis by the image processing algorithms. These subsystems are show schematically in the Figure 2.

However, each one of the IFS equipments must be configured for the correct capture and the correct formation of the frame. Usually, in less automated systems, its configuration is done by the experimenter, who, for example, determines the increase of the laser intensity and/or the shutter time of the camera when he verifies that the image sent to the IPS does not have enough bright and contrast. There is a feedback in this process: the equipments must be continuously regulated by the analysis of the captured images. This action of the IPS on the IFS is symbolized in Figure 2 through a control signal $S_c(t)$.

The image signal $S_I(t)$ outputted by the camera, the camera frame rate T_q and the scale factor FE are sent to IPS. The scale factor FE relates the coordinates of the particle ensemble (object domain, perspective, or scene) with the coordinates of its images (image domain, projection, or image) [15]. The image signal $S_I(t)$ and the parameters T_q and FE are used in the IPS by its image processing algorithms for the calculation of the velocity field [12].

Velocimetry System through Image Processing

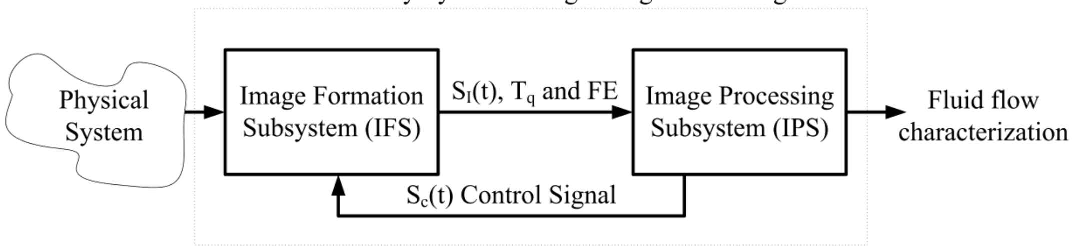


Figure 2 – A velocimetry system through image processing.

It should be noted that the unique information that the IPS needs is the image signal $S_I(t)$ and the parameters T_q and FE, which completely characterize the physical system (fluid flow in this case) and the IFS subsystem, respectively. That is, for the IPS, it is not important the nature of the physical system, not even which are the equipments employed in the IFS, but just the knowledge of the 3 previous commented quantities: $S_I(t)$, T_q, and FE . Therewith, it is possible to compute the velocity field using the developed algorithms.

The abstraction level that was used here to describe the velocimetry system (through just 2 subsystems) is important, because it propitiates that new algorithms can be developed, independently of the type of fluid flow in which these algorithms are applied and of the type of equipments employed in the IFS.

The Figure 3 illustrates the situation where a particle ensemble follows a streamline of an arbitrary fluid flow. The ensemble of particles passes by the arbitrary points P_0, P_1 e P_2 in also arbitrary times given by t_0, t_1 e t_2, respectively. Considering $\Delta t_j \approx$ constant $= \Delta t = 1/T_q$, the time-mean velocity of this particle ensemble for a time of $J\cdot\Delta t$ is given by the following equation:

$$\left\langle \mathbf{v} \right\rangle_o = \frac{\sum_{j=1}^{J} \Delta \mathbf{r}_j}{\sum_{j=1}^{J} \Delta t_j} = \frac{1}{J} \cdot \sum_{j=1}^{J} \mathbf{v}_j \qquad (1)$$

Where j is the time index, $\mathbf{r} = x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z$, and ($\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$) are the unitary vectors in the (x,y,z) directions, respectively. For j=1, the equation (1) gives the instantaneous velocity vector of the particle ensemble.

The transformation of a perspective in a projection is a nonlinear operation where one of the spatial dimensions is eliminated. When the interest falls in dynamical quantities like the velocity, it is observed mathematically that there is some influences of this eliminated velocity component in the computation of the other ones. However, in certain situations, these influences may be neglected and this transformation becomes linear. Then, the object velocity can be obtained from the image velocity by just the multiplication of the constant FE [12]. In these circumstances:

$$\left\langle \mathbf{v} \right\rangle_o \approx FE \cdot \left\langle \mathbf{v} \right\rangle_i = FE \cdot T_q \sum_{\alpha=\{x,y,z\}} \left\langle \Delta r \right\rangle_{i\alpha} \mathbf{e}_\alpha \qquad (2)$$

Where $\left\langle \Delta \mathbf{r} \right\rangle_i$ is the average displacement of the particle ensemble in the captured frames along the time. In the area of image processing, the particle ensemble can be interpreted as just a particle pattern, which must be tracked in the sequence of frames captured from the fluid flow. This nomenclature will be used in the remaining text of this paper.

Nevertheless, in the region of the fluid flow captured by the electronic image sensor, there are innumerous particle patterns that should be tracked. Supposing that these patterns do not interact with each other, it is possible to analyze its dynamics in the same way as in the Figure 3. Then, the results of equation (2) can be extended for each one of these patterns, just letting that the displacement $\left\langle \Delta \mathbf{r} \right\rangle_i$ becomes a function of the spatial coordinates (x,y,z), and therewith acquiring the characteristics of a field quantity. How it was previously commented, due to restrictions of our instrumentation system, the IPVFlow V1.0 tool is able just to compute 2 spatial dimensions of the velocity field. Then, equation (2) is restricted to compute just α={x,y} velocity components. This is not critical if the frame-rate T_q is sufficiently large such that the particle pattern does not significantly change between two successive frames, what means that the particles in this interval of time do not leave the frame plane.

Since some aspects about the principle of velocimetry have already been elucidated, in the next sections it will be presented the main algorithm that runs inside of the IPVFlow V1.0, the computational tool developed by us to measure the two-dimensional velocity fields of fluid flows.

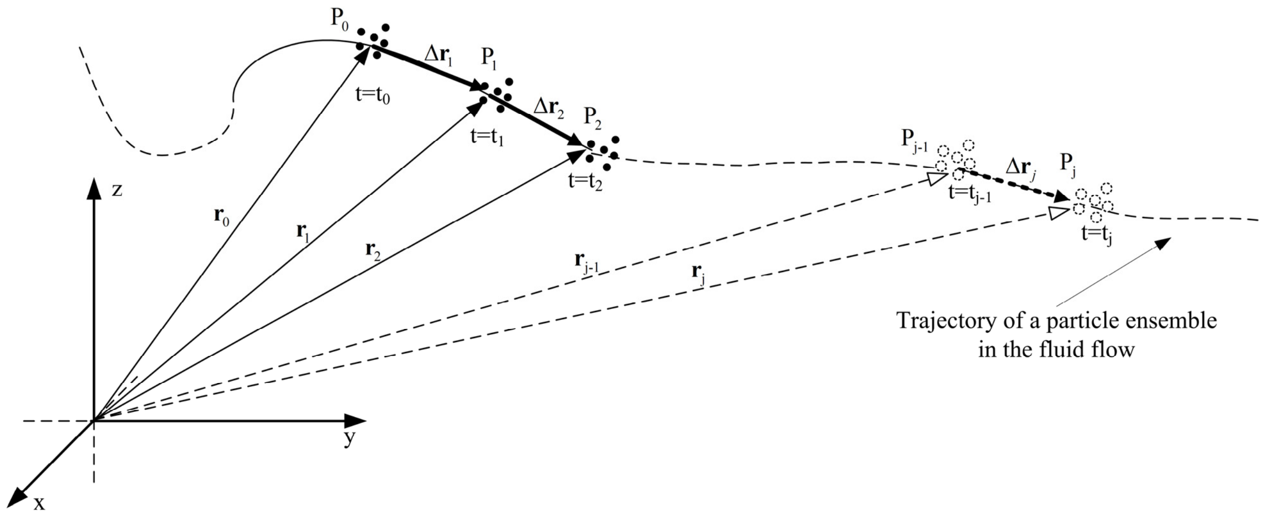


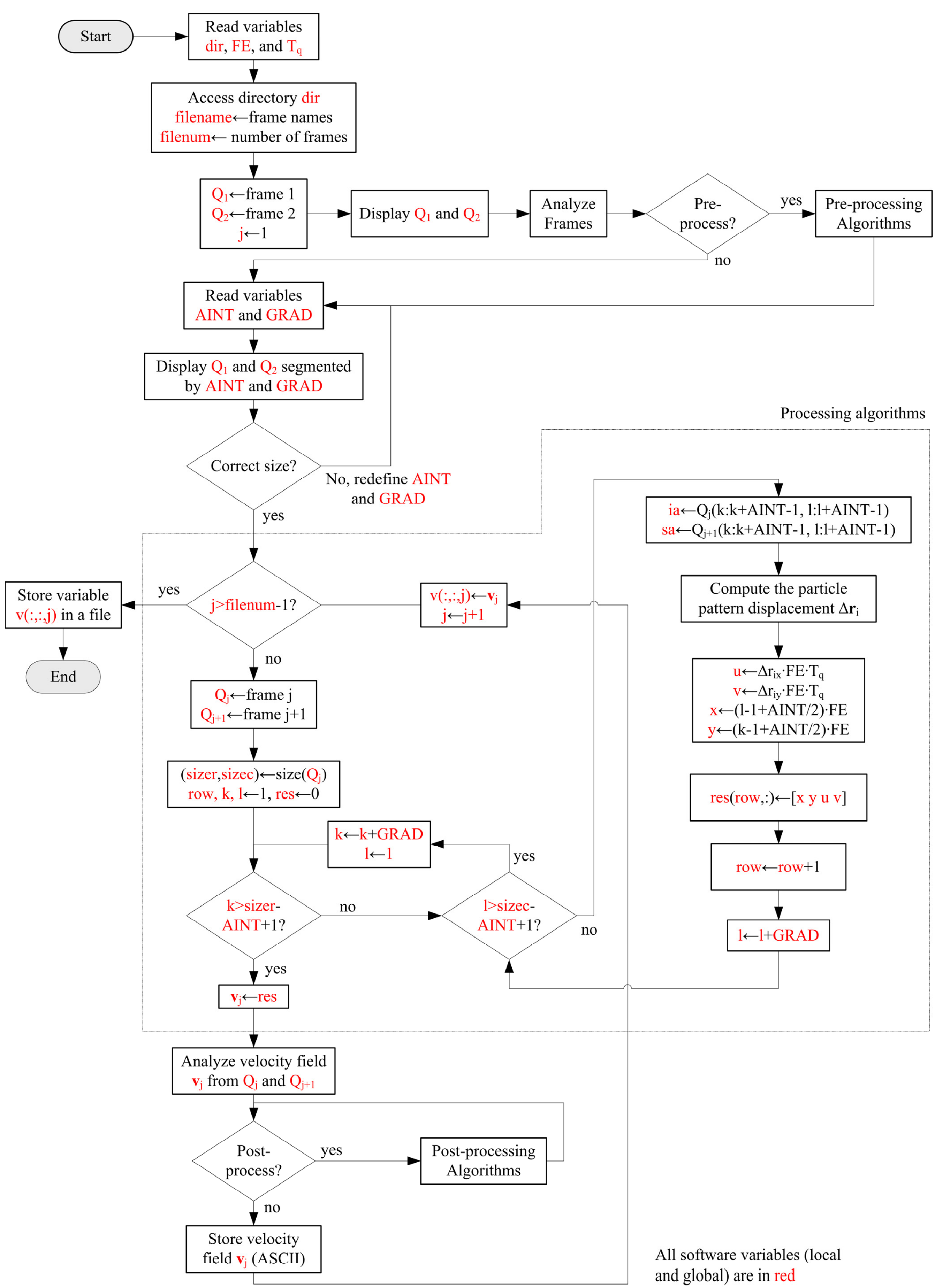
Figure 3 – An arbitrary trajectory of a particle ensemble.

3 The main algorithm

The main algorithm that runs inside the IPVFlow V1.0 is represented through blocks diagram in the Figure 4. It was used the ANSI standard in this representation and all software variables (local and global) are showed in red. These variables can be scalars or vectorial quantities. Each dimension of a vectorial quantity is specified after the variable name inside parentheses. When referring for one dimension, the others are filled by ":". An example: the second velocity field captured from the fluid flow, a two-dimensional quantity, is represented by $\mathbf{v}(:,:,2)$.

Principally, the IPVFlow V1.0 is composed by pre-processing, processing and post-processing blocks and logical structures that interconnect them [12]. How commented before, the objective here is to show how these previous processing blocks interact. Detailed explanation about the overall system is found in reference [12]. The algorithms for calculation of the displacement $\Delta \mathbf{r}_i$ are explained in [17], while the remaining will be subject of future publications.

The analog signal $S_I(t)$ originated from the IFS subsystem of Figure 2 is transformed into bitmap frames through a frame grabber connected to a PC and stored in a directory, previous defined by the user. These frames captured from the fluid flow along the time are represented by $\{Q_j\ Q_{j-1}\ Q_{j-2}\ ...\ Q_0\}$, where j indexes the time evolution in the same way as in equation (1).



All software variables (local and global) are in red

Figure 4 – The main algorithm of IPVFlow V1.0 software.

The input variables of the IPVFlow V1.0 software are the scale factor "FE", the frame-rate "T_q", and the directory "dir", where the frame sequence {Q_j Q_{j-1} Q_{j-2} ... Q_0} are stored. Then, the total number of frames and its filenames are computed and stored, respectively, in the variables "filenum" and "filename".

The first, j=1, and the second, j=2, frame captured from the fluid flow are loaded in variables named as "Q_1" and "Q_2", respectively, and displayed. Algorithms are executed and it is analyzed if it is necessary to pre-process the fluid flow frames. If it was, then all frames are pre-processed and stored again with the same filenames given by the "filename" array. If it was not necessary, then the user (in this case, the experimentalist) is asked to insert the variables "AINT" and "GRAD", for the particle patterns segmentation. The pre-processing algorithms consist of noise filters, image segmentation, image enhancement, image restoration, and others [12, 14, 17].

The variable "AINT" specify the size in pixels of the particle pattern that is selected in the sequence of frames {Q_j Q_{j-1} Q_{j-2} ... Q_0}. The distance between the two consecutively selected particle patterns is given by "GRAD". In the case of GRAD<AINT, there will be an overlapping between the particle patterns. That is, it will have particles that will be common in neighbor particle patterns selected in the frames. This mesh established by "AINT" and "GRAD" are plotted over the frames "Q_1" and "Q_2". Thus, by visual inspection, the user can decide if it is necessary to redefine these variables.

The main loop of the software consists in loading the frames Q_j and Q_{j-1}, in storing its number of rows in variable "sizer", in storing its number of columns in the variable "sizec', in segmenting them in particle patterns of "AINT" pixels with distance of "GRAD" pixels, and in calculating its velocity vector $\mathbf{v}$. The variable "j" is increased until it reaches the total number of frames "filenum" less 1. It means that, for "filenum" frames, there will be "filenum"-1 velocity fields.

The variables "k" and "l" index, respectively, the number of rows and columns of each frame. These variables can vary from 1 to "sizer-AINT+1", for the case of "k", or from 1 to "sizec-AINT+1", for the case of "l". The "k" and "l" variables are increased by the amount of "GRAD", until they reach the frame dimensions.

The process here to compute the velocity field is simple. For a fixed row "k", many particle patterns of size AINT are selected through the increasing of the columns "l" by an amount of "GRAD". The particle patterns selected in the "Q_j" and "Q_{j-1}" are designated by the variables "ia" and "sa", respectively. From "ia" and "sa", the particle pattern is tracked and its two dimensional displacement $\Delta\mathbf{r}_i$ is computed. Through the multiplication by just FE and T_q, the velocity vector for this particle pattern is calculated. For each value of "k", the results for all possible values of "l", together with its spatial coordinate (x,y), are stored in the variable "res" as a row vector, indexed by the variable "row". When the variable "k" reaches its maximum value, then the variable "res" will contain the entire velocity field for an instant of time indexed by "j".

After the calculation of the velocity field for a given "j" $\mathbf{v}_j$, it is verified if it is necessary to post-process this velocity field. The post-processing algorithms are applied to validate $\mathbf{v}_j$, eliminating the spurious velocity vector and replacing them by interpolation of its neighbors' [12, 17]. The resultant instantaneous velocity field $\mathbf{v}_j$ are saved as an ASCII file, saved in another three-dimensional variable "$\mathbf{v}$(:,:,j)", and plotted over the frame Q_j, which is also saved as a bitmap image. Finally, the variable "j" is incremented and, if it reaches the value "filenum", the main loop is finalized and the three-dimensional variable "$\mathbf{v}$(:,:,j)" is stored. Otherwise, new frames Q_j and Q_{j-1} are loaded and its velocity field is calculated, performing again the same procedure explained in this section.

Note that the variable "$\mathbf{v}$(:,:,j)" contains all spatial and temporal information about the fluid flow. It is suitable to store this in a single variable since new quantities pertaining to the fluid flow are expected to be calculated, or the previously obtained results are expected to be used for composing new graphics, animations or videos.

All the software algorithms were implemented in the MATLAB environment. MATLAB is a high performance language for technical computing suitable for fast prototyping and for this reason was chosen here.

4 Discussion and results

For demonstrating the functionality of the IPVFlow V1.0 software, this computational tool was applied in three different physical systems. Some input variables like "AINT" and "GRAD" of the software were changed to exemplify its operation. The computed velocity fields will be shown as vector graphics over the fluid flow frames.

In the Figure 5, it is presented the instantaneous velocity field for a wall shear flow [18]. The frames have a size of 256x256 pixels and were segmented in square particle patterns of AINT=16 pixels, with GRAD=16 pixels. The frames Q_j and Q_{j-1} are shown respectively in Figure 5a and in Figure 5b. The total number of particles is 4000, with a mean particle diameter of 5 pixels. The calculated velocity field is presented in blue in the Figure 5c.

The second physical system is a jet impingement flow [18]. The frames have a size of 256x256 pixels and were segmented in square particle patterns of AINT=16 pixels, but, differently from the previously fluid flow, it was used GRAD=8 pixels. As a consequence, it can be seen clearly the improvement of the spatial resolution of the velocity field. The total number of particles is 12000, with a mean particle diameter of 7 pixels. The frames Q_j and Q_{j-1} are shown respectively in Figure 6a and in Figure 6b. Its velocity field is shown in red in Figure 6c.

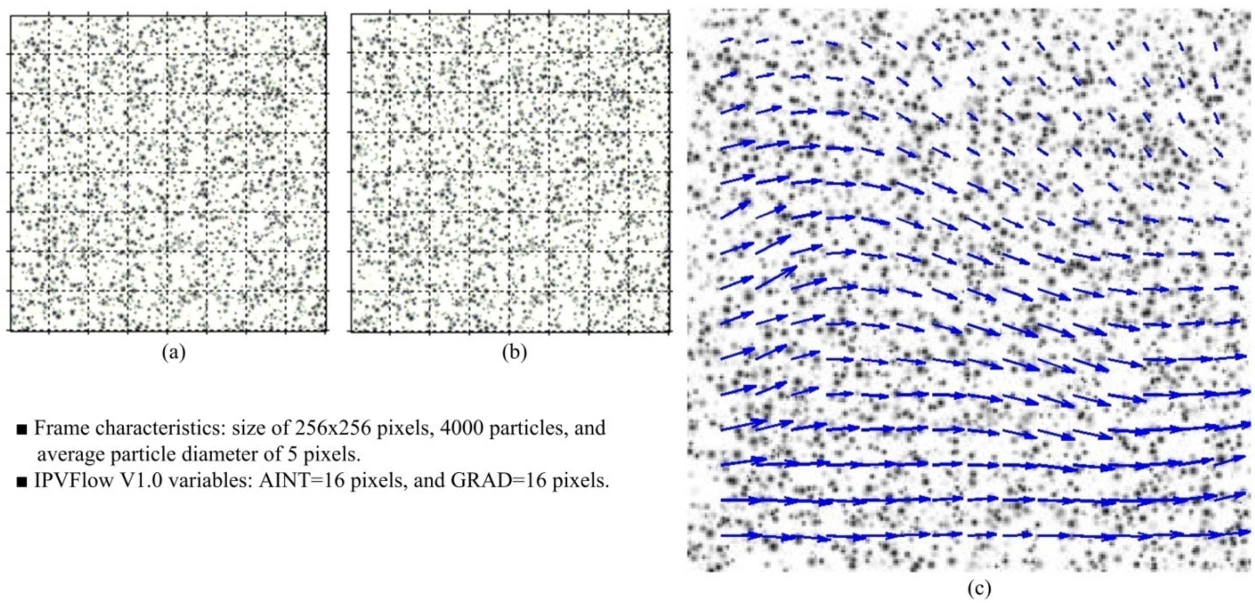


Figure 5 – Wall shear flow: (a) Frame Q_j, (b) Frame Q_{j-1}, and (c) Instantaneous velocity field.

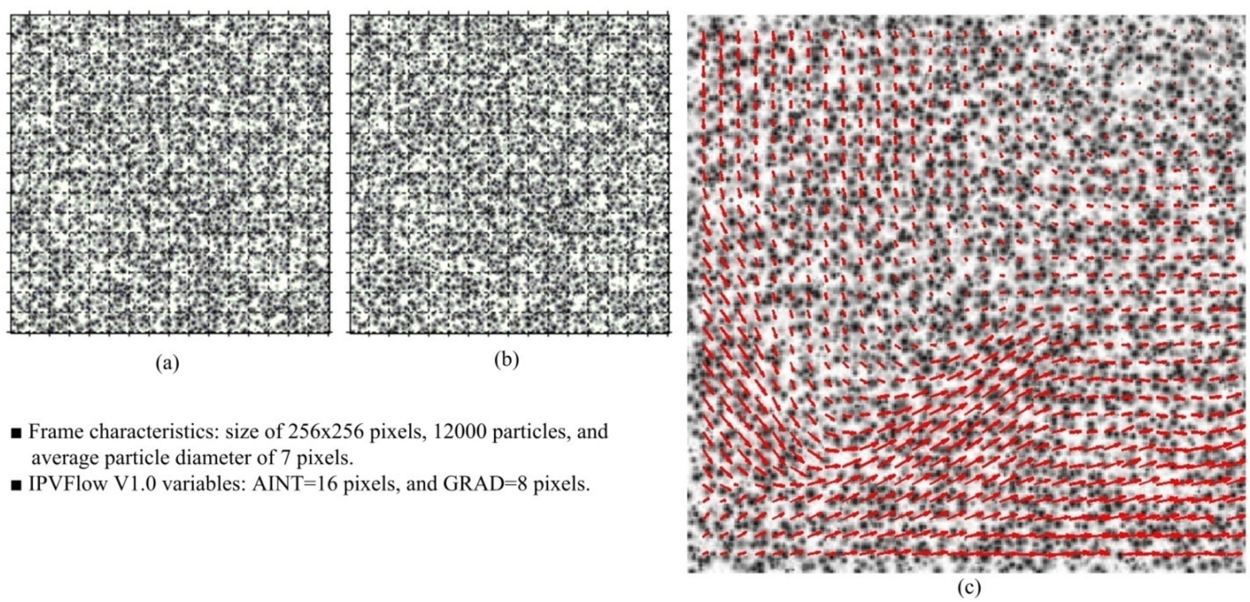


Figure 6 – Jet impingement flow: (a) Frame Q_j, (b) Frame Q_{j-1}, and (c) Instantaneous velocity field.

The third and last physical system analyzed here is a flow internal to a metallurgical ladle, with an elliptical transversal section. These experiments have been originally carried out by us to study which is the influence of the velocity field structure in the mixing characteristics of the steel, which is agitated inside the ladle. Further information can be found in references [7] [19].

The frames have a size of 640x480 pixels and were segmented in square particle patterns of AINT=32 pixels, with distance between them of GRAD=8 pixels. The total number of particles is 24000, with a mean particle diameter of 3 pixels. The frames Q_j and Q_{j-1} are shown respectively in Figure 7a and in Figure 7b. Differently of the two previous physical systems, instead of the instantaneous velocity field, the quantity calculated here is the time-mean velocity field. For that, equation (1) was applied for J=360 frames, totalizing an average process of 12s, since the employed frame-rate was 30Q/s. This time-mean velocity field is shown in yellow in Figure 7c.

It can be seen in this figure, two symmetrical recirculation regions of the liquid close to the ladle surface, and in both sides of the ladle wall. The two vortexes present opposite rotation directions, originating a high rising fluid flow in the central region and low velocity regions, the so called "dead zones", in the ladle bottom. These "dead zones" degrade the mixing quality of the steel inside the ladle and should be avoided [19].

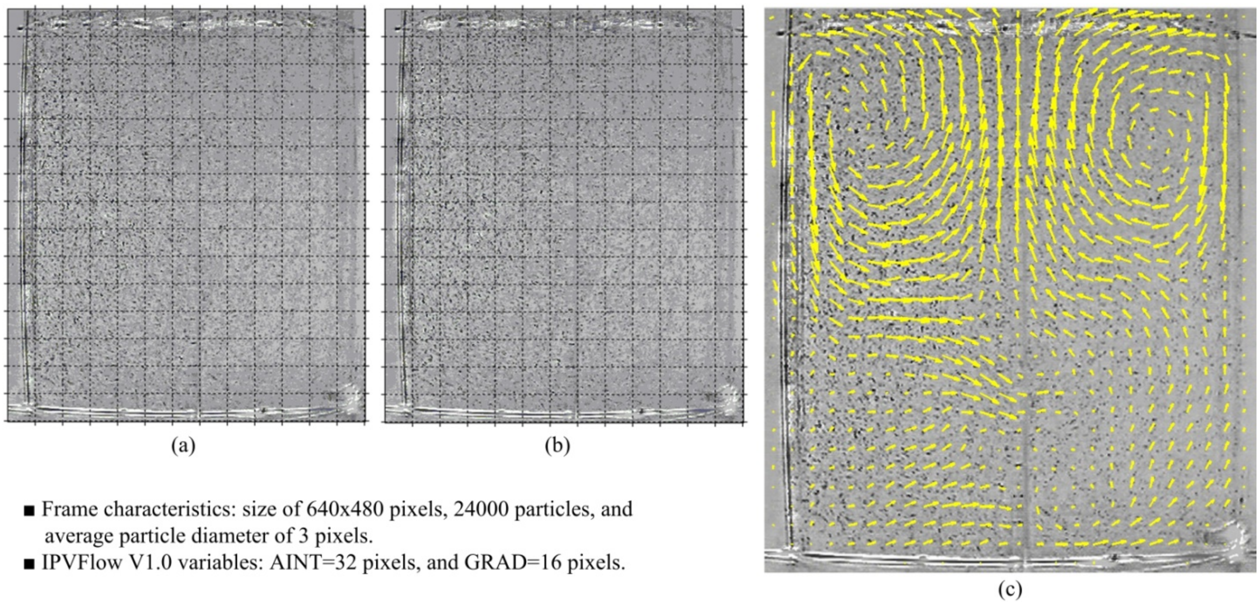


Figure 7 – Fluid flow inside a metallurgical ladle: (a) Frame Q_j, (b) Frame Q_{j-1}, and (c) Time-mean velocity field for 360 frames.

5 Conclusions

The techniques involving image processing for the first time offered experimentalists in the area of fluid dynamics the ability to measure the instantaneous spatial nature of a fluid flow. With this, the evolution of the velocity field of the fluid flow as a function of time became possible. Thus, the detailed temporal statistics obtained globally from these measurements propitiated that unique information about the fluid flow dynamics, fluid flow transport and the motion of structures in the fluid flow have been quantified.

A velocimetry system through image processing is complex. This involves physical concepts about fluid dynamics, instrumentation techniques and theory of signal processing. In this way, in this work, it was just presented the main algorithm that governs the IPVFlow V1.0 software, a computational tool developed by us to compose the IPS subsystem and which was explained in section 2 of this paper.

The division of the velocimetry system in two functionally distinct subsystems, namely the image formation subsystem (IFS) and image processing subsystem (IPS), helps to understanding how a computational tool for velocity field measurement of fluid flows is projected, implemented, and can be changed to improve the algorithms performance.

The algorithms showed in this work were implemented in the MATLAB environment. For demonstrating its functionality, the tool was applied in 3 distinct fluid flows: a wall shear flow, a jet impingement flow, and a flow inside a metallurgical ladle. From these, different quantities pertaining to the fluid flow as the instantaneous and the time-mean velocity field were shown.

Future works will include new algorithms to calculate other quantities of practical interest in fluid mechanics like vorticity, intensity of turbulence, circulation, pressure, and others. To improve the portability of the IPVFlow V1.0 tool, it is intended to implement the algorithms of IPS also in C++ language.

To conclude, the algorithms presented satisfactory results. They are a part of an instrumentation system that has an uncertain in the measured velocity vectors less than 1.6%.

6 Acknowledgements

The author Roger Pizzato Nunes acknowledges the financial support from CNPq, Brazil.

7 References

[1] Landau, L. D., Lifshitz, E. M., *Fluid Mechanics*, Volume 6 of Course of Theoretical Physics, Addison-Wesley, London, 1959.

[2] Thomas, E., Watson, M., “First experiments in the Dusty Plasma Experiment device”, *Physics of Plasmas*, 1999, vol. 6, pp. 4111-4117.

[3] Schlichting, H., *Boundary-Layer Theory*, McGraw-Hill, New York. 1968.

[4] Yildirim, I., Cetiner, O., Unal, M. F., “PIV Measurements of the wake interactions for a Circular Cylinder behind an Airfoil”, *International Symposium on Particle Image Velocimetry*, 2005.

[5] Chow, Y. C., Katz, J., Uzol, O., Meneveau, C., “An Investigation of Axial Turbomachinery Flows Using PIV in an Optically-Unobstructed Facility”, *International Symposium on Transport Phenomena and Dynamics of Rotating Machinery*, 2002.

[6] Vieira, E. D. R., Dall’Agnol, E., Mansur, S. S., “Flow visualization of heart valves prostheses in a steady flow model”, *Revista Brasileira de Engenharia Biomédica*, 1999, vol. 15, pp. 63-68.

[7] Nunes, R. P., Morales, J. A., Vilela, A. C. F., van der Laan, F. T., “Analysis of the Fluid Flow Structure in a Physical Model of a Steelmaking Ladle Through Image Processing Velocimetry”, *12th International Symposium on Flow Visualization*, September 10-14, 2006, German Aerospace Center (DLR), Göttingen, Germany.

[8] Gioia, G., Chakraborty, O., “Turbulent Friction in Rough Pipes and the Energy Spectrum of the Phenomenological Theory”, *Physical Review Letters*, 2006, pp. 1-4.

[9] Goldenfeld, N., “Roughness-Induced Critical Phenomena in a Turbulent Flow”, *Physical Review Letters*, 2006, pp. 1-4.

[10] Doebelin, E. O., *Measurement Systems Application and Design*, McGraw-Hill, New York. 1990.

[11] Lai, W. T., Bjorkquist, D. C., Abbott, M. P., Naqwi, A. A.,“Video systems for PIV recording”, *Journal of Measurement Science and Technology*, 1998, vol. 9, pp. 297-308.

[12] Nunes, R. P., *Projeto e Implementação de um Sistema de Instrumentação Eletro-Eletrônica para Caracterização de Escoamentos através de Processamento Digital de Imagens*, Dissertação de Mestrado, Departamento de Engenharia Mecânica, Escola de Engenharia, UFRGS, 2005.

[13] Shames, I. H., *Mechanics of Fluids*, McGraw-Hill, New York, 1992.

[14] Gonzalez, R. C., Woods, R. E., *Processamento de Imagens Digitais*, Edgard Blücher, São Paulo, 2002.

[15] Schalkoff, R. J., *Digital Image Processing and Computer Vision*, John Wiley & Sons, Canada, 1989.

[16] Nunes, R. P., van der Laan, F. T., “Velocity field measurement for a high turbulent duct flow through digital image processing”, *11th Brazilian Congress of Thermal Sciences and Engineering*, December 5-8, 2006.

[17] Nunes, R. P., van der Laan, F. T., “Design and implementation of an image processing algorithm for velocimetry systems”, *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2007, submitted.

[18] Okamoto, K., Nishio, S., Saga, T. and Kobayashi, T., "Standard images for particle-image velocimetry," *Measurement Science and Technology*, 2000, vol. 11, pp. 685-691.

[19] Nunes, R. P., Morales, J. A., van der Laan, F. T., “Characterization of the Dead Zones in a Physical Model of a Steelmaking Ladle Through Particle Image Velocimetry”, *IEEE Transactions on Instrumentation and Measurement*, 2007, submitted.