Development of a 3D Particle Tracking Velocimetry method and its associated Python-coded software for image processing

Kevin Laurent[1], Magali Barthès[1,*], V. Lepiller[1], Y. Bailly[1]

[1]FEMTO-ST institute, UMR CNRS 6174, Energie Department, University of Franche-Comté, Belfort, France
*corresponding author: magali.barthes@femto-st.fr

Abstract The aim of this experimental study was to use classical 2D PTV (Particle Tracking Velocimetry), and to extend it to develop a 3D method. One of PTV method advantages is the study of a situated phenomenon thanks to its Lagrangian specification. Thus, those 2D and 3D methods can be used to investigate simple, as well as more complex flows. For the 2D method, a plan of the seeded flow was illuminated by a monochromatic source (Nd-YAG laser) which reflects on tracer particles, and a camera recorded images of the flow. Nevertheless, most of flows correspond to 3D non-stationary phenomena. Thus the objective was to develop an optical method, with its image processing software, to be able to obtain a 3 dimensions and 3 velocity components description of the phenomena. The optical method we developed is based on both PTV and Rainbow Volumic Velocimetry (RVV) principles. It consisted in illuminating a volume of seeded flow with a continuous polychromatic spectrum, obtained from a white light beam dispersion using a blazed reflecting grating. The color enabled us to determine the third spatial component, *i.e.* the displacement in the depth direction (perpendicular to the observation window). Color images were captured with a 3CCD camera. To process these images and analyze the flow (analysis of tracer's colors in the polychromatic volume), we developed a computer program using Python™ that we called "PyTV". This software enables us to process images from classical 2D PTV method, as well as from the 3D method. 2D or 3D velocity vector fields were obtained. The image processing code was validated using synthetic images we generated with a well-known 2D or 3D displacement. The software processing was found to have a good accuracy (error under 5%).
Keywords: 3D PTV, RVV, Laser, Velocimetry

1 Introduction

Some optical methods enable to study space and temporal displacement of each element of fluid flow in a (quasi) non-invasive way. In literature, two types of optical method [1] can be found: the first one is based on fluid properties variation like, for instance, shadowgraphs [2] whose principle is based on the refractive index variation according to the fluid temperature variation. The second type of method is based on light diffusion on markers like, for instance, PIV (Particle Image Velocimetry) which is an accurate and quantitative method used for measuring large instantaneous fluid velocity fields with high spatial resolution. Standard PIV (two dimensions 2D, two velocity vectors 2C) extracts the displacements as functions of the in-plane directions [3]. All the optical methods have advantages and drawbacks, an overview of some of them, with the space and temporal information that can be obtained, is given on Fig. 1. Nowadays, most of "classical" optical methods, such as interferometry, Schlieren, shadowgraph, PIV, LIF (Laser Induced Fluorescence), PTV (Particle Tracking Velocimetry), tomography etc. delivered only a 2D information. Thus, 3D information is usually reconstructed by assembling "plane by plane" 2D results [4]. Even dual Plane Stereoscopic PIV (2D-3C) only allows the determination of the whole fluid velocity gradient tensor in the investigated plane (2D). Non-intrusive 3D measurement, like holography [5], are still hard to set up (and sometimes do not enable the study of non-stationary phenomenon).
Another method, which is 3D-3C and was developed in our institute, is the RVV (Rainbow Volumic Velocimetry) one [6][7]. Instead of illuminate a plane of the flow with a monochromatic light (laser) sheet, this method is based on the illumination of a volume of the seeded flow with a continuous polychromatic spectrum. The color is used to determine the third component. The polychromatic spectrum is obtained from a white light beam dispersion using a blazed reflecting grating and a polychromatic volume of light is obtained thanks to optical components. Each particle can reflect a color, recorded with a high exposition time by a 3CCD camera. Thus, for each particle, a streak corresponding to the displacement of the particle is obtained. From position and color of both ends of each mark, and knowing the exposition time, particle displacements and velocities can be determined. This method have already been checked [8] and used, for instance, on flows downstream an obstacle [9]. This method requires a low seeding density. To improve the 3D velocity

determination (for instance with rotational flows), our objective was to combine PTV and RVV methods. Classical PTV method [10] enables to determine velocity by following the particle displacement between consecutive images. Since each particle is individually tracked, like Lagrangian method, contrary to the PIV method, based on correlation calculation between two areas of images which contain groups of particles [11]. Classical PTV is based on (as for PIV) a seeded flow (lower seeding density than in PIV) illuminated with a laser sheet (monochromatic). The advantage of PTV is then to locally describe phenomena in flow, but this method remains a 2D one.

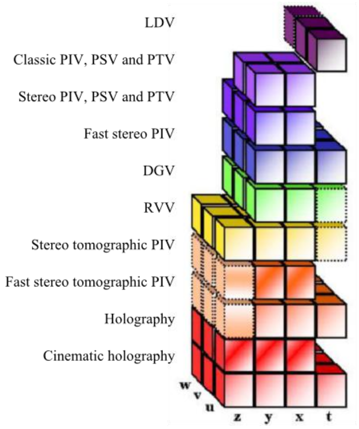

Fig. 1 An overview of the existing optical velocimetry method [12]

2 Experimental setup: 2D and 3D PTV

As previously mentioned, our objective was to develop a 3D-3C PTV method based on “classical” PTV and RVV principles.
The 2D classical method requires a seeded flow which is illuminated with a laser sheet. This last one is created from a laser beam which is split into a sheet thanks to optical components (lenses).
Concerning the 3D method, the experimental setup is given on the Fig. 2a. The light source we used is a white laser (Koheras SuperK Versa). At the maximum of its power, only approximatively 9.3% of the spectrum is in visible wavelength: the major part of the white laser wavelengths correspond to infrared ones. Yet, those infrared induced high energy input in the system, and can induced thermal effect which could be responsible of flow structure modification. Thus, to avoid the influence of energy related to infrared, two dielectric mirror (M1 and M2 on Fig. 2a) were used: 99.5% of the light reflected by the mirror are in visible spectrum. Then, the laser beam was diffracted into a rainbow thanks to a blazed grating (Fig. 2b). The only diffraction order we used was the first one: the 0-order had the highest energy but was not diffracted enough (it was just the image of the laser source), and the other orders had too low energy levels. Thus, only the first order was then split into a volume of light thanks to a plano-convex lens.

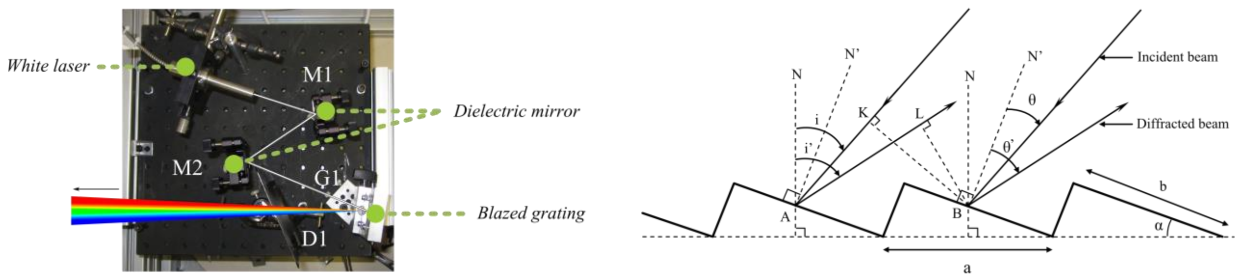

Fig. 2 (a) Optic device; (b) Blazed grating principle [13]

The volume of rainbow lights (continuum spectrum) illuminates a seeded flow, and a 3CCD camera must be used to record images of the particles. On the 2D images, each particle has a spatial position (x, y). Then, the color of each particle corresponds to the position of the particle in the last spatial component: the depth, z. In addition to the classical spatial calibration in (x, y) direction, a color calibration is necessary to obtain the position of the particle in depth. On the image, the color of each particle corresponds to values on each three color channels of the camera (Red, Green and Blue). The color calibration must link the particle color (image information) to the spatial position along the z axis (Fig. 3).

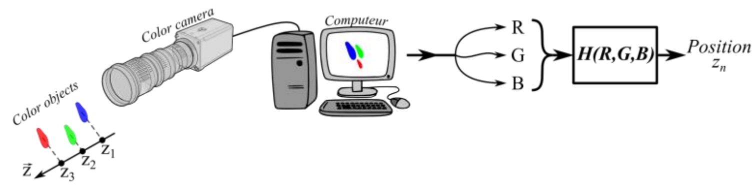

Fig. 3 Color using in PTV

3 Image processing software

To obtain 2D or 3D velocity vector fields, the images acquired with the camera need to be processed. We developed a Python-coded program which is able to identify and track particles in a succession of consecutive images. Python™ is a programming language, object-oriented, open-source and cross-platform. The used scientific libraries for our process were: PIL (Python Imaging Library), SciPy (Scientist Python) and NumPy (Numerical Python). The developed software, called *"PyTV"*, is able to process classical 2D PTV as well as 3D one. The program structure is composed of three types of module: main (bold boxes on the Fig. 4a), process (italic text on the Fig. 4a) and minor (dotted line on Fig. 4a). All setting for the image processing are inputted *via* the graphic user interface corresponding to the *"PyTV"* module. This last one sets off the *"mainprocess"* module which is the only one being linked to all modules: so, it handles both execution order and communication between each process module. The *"mainprocess"* algorithm is given on Fig. 4b.

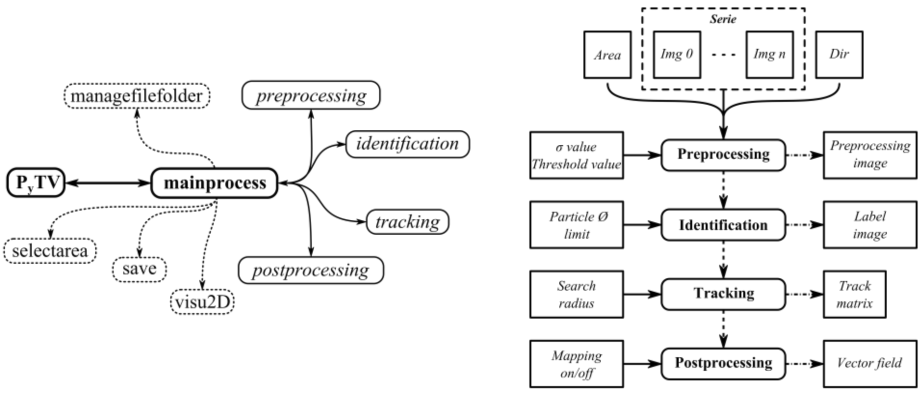

Fig. 4 Diagram of (a) the software modules and of (b) the main algorithm

This *"mainprocess"* algorithm is divided into four steps, each one corresponding to a specific process module. The first step corresponds to the *"Preprocessing"* one: raw images quality is improved to enable a better identification of particles. To do so, numerous images are recorded before running the experiment. From those images, a mean image is computed. To remove the fixed element (dead pixel or steady reflection) from images, and also to reduce images noise (due to the camera), this mean image is subtracted to each image acquired during the experiment. Lastly, a Gaussian filter and a thresholding are applied to the recorded images.

After the "preprocessing" module, the "identification" enables to index each particle in each image. To identify particles, we first convert the previous preprocessed images into binary ones. Each detected object is identified and labelled in the image. Then, to remove objects that are not a particle, we developed a Gaussian fitting. Thanks to this fitting, six parameters $(x_0, y_0, \sigma_x, \sigma_y, \theta, M)$ are extracted from the 2D Gaussian function (1)[14]:

$$g(x, y, x_0, y_0, \sigma_x, \sigma_y, \theta, M) = Me^{-8[A(x-x_0)^2+2B(x-x_0)(y-y_0)+C(y-y_0)^2]} \quad (1)$$

With A, B and C coefficients defined in equation (2):

$$\begin{cases} A = \dfrac{\cos^2\theta}{\sigma_x^2} + \dfrac{\sin^2\theta}{\sigma_y^2} \\ B = \dfrac{\cos 2\theta}{2\sigma_x^2} + \dfrac{\sin 2\theta}{2\sigma_y^2} \\ C = \dfrac{\sin^2\theta}{\sigma_x^2} + \dfrac{\cos^2\theta}{\sigma_y^2} \end{cases} \tag{2}$$

An example of a required criterion for an object to be a particle is that the σ ratio, given on equation (3), must be between *LGRP* (Low Gaussian Ratio Parameter) and *HGRP* (High Gaussian Ratio Parameter) values:

$$LGRP \geq \frac{\sigma_x}{\sigma_y} \geq HGRP \tag{3}$$

With $LGFP \in]0,1]$ and $HGFP \in [1,\infty[$, both values are given as input in the program by the user. Another example of filtering criterion is the value of the Gaussian amplitude "M" (1) which mustn't be too high with regard to the image color depth. The Gaussian fitting also enables a better localization of the particle in the image, since it allows to obtain subpixel coordinates (x_0 and y_0): displacements determination (and so velocities) will be more accurate. The next step after this "*identification*" module is the tracking of all particles. In the "*tracking*" module (Fig. 5), we developed a sequential tracking, *i.e.* each particle will be followed during the whole images series (as long as the particle is still in the image).

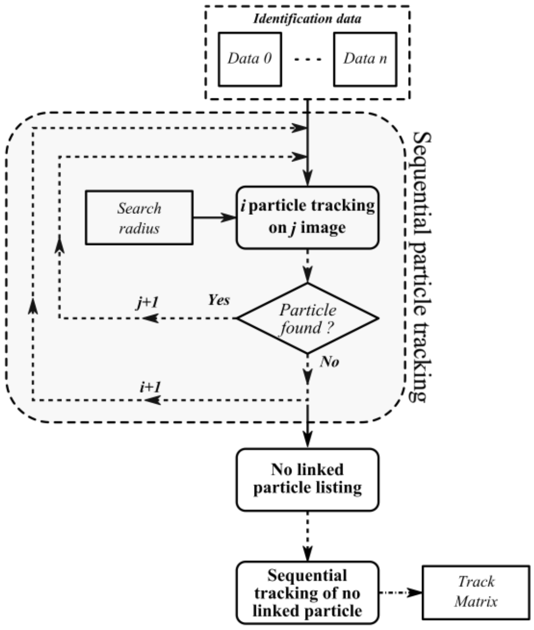

Fig. 5 Tracking algorithm

All the tracking data are gathered together in a matrix: its rows correspond to the particle track, and its columns to the image number. It is thus possible to follow only one particle during the whole process (trajectory of one particle along image series).

Lastly, after the "*tracking*" process, the last step corresponds to the post-processing one. Thanks to identification (subpixel position of each labelled particle) and tracking data (tracking matrix), we can compute the particle displacement (in pixels). Apart from the tracking which becomes 3D for the 3D PTV, the main difference between the 2D and 3D PTV is the calibration to convert pixels into length. Whereas for the 2D PTV, only a spatial calibration is required (in the x, y plan), for the 3D one, a colorimetric calibration must be done. Indeed, information obtained from the 3CCD camera, *i.e.* RGB channels, must be converted into spatial position. We need to obtain the camera description (Red, Green, Blue) of the white source spectrum along the z axis (depth, Fig. 3). The spectrum of the white laser (Koheras SuperK Versa) obtained on the 3-CCD color camera (Sony DXC-990P) is given on Fig. 6a. To obtain a correct RGB signal, with a reduced noise (camera) the image of the spectrum on Fig. 6a was pre-processed. To do so, a mean images of hundreds of snapshots with the lens cover on was removed from the raw image of the spectrum. The snapshots with the lens cap on were taken before and after the experiment, to take into account the effect of thermal noise. The corresponding RGB level (determined along the horizontal axis of the spectrum) is showed on Fig. 6b and still exhibits some noise. The signal was thus processed with a Fast Fourier Transform (FFT) algorithm to extract frequencies of our spectrum. Then we applied a low-pass filter to remove all high frequencies (which corresponded to the noise). Then we got back to our signal thanks to an Inverse Fast Fourier Transform (IFFT) algorithm. The smoothed signal can be seen on the Fig. 6c.

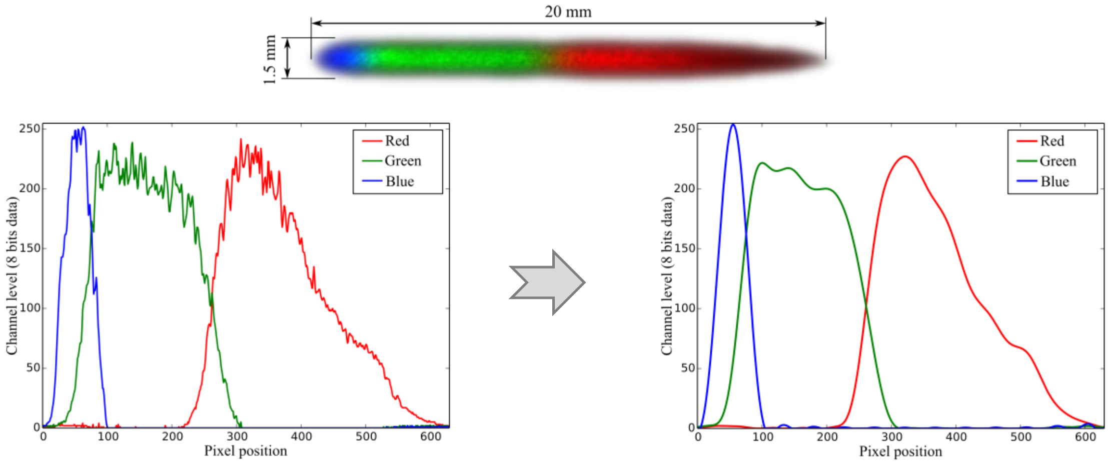

Fig. 6 (a) White laser diffracted spectrum (laser power: 95%);
(b) Corresponding color channels along horizontal level; (c) FFT process result on Fig. 6b

On the Fig. 6, it is obvious that the blue represents only a little part of the spectrum: indeed, blue spatially depict only 16% of our spectrum. Once the spatial and colorimetric calibration done, the displacement vector fields can be obtained. Knowing the time between images, velocity vector field can be computed. The last steps of the post-processing module correspond to the filtering of false vectors, and to the possibility to interpolate (or not) the vector field.

4 Validation

To validate our software, we generated synthetic images allowing us to configure many parameters on image like its format, its color depth, its size or its signal noise ratio (SNR), and on particles like their sizes, their color limits, their behaviors (translation, rotation).
Figure 7 represents the generation of two synthetic images whose particles have a displacement only along z axis (Fig. 7). The images have 512 by 512 pixel size, and contain one hundred particles whose diameters vary between seven and eight pixel.

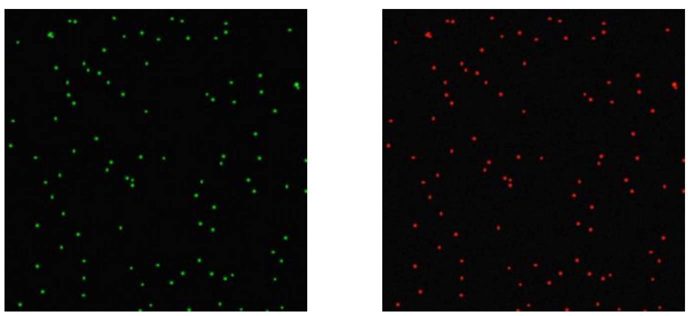

Fig. 7 Two synthetic images with displacement along z axis

These synthetic images (Fig. 8) have been processed by our software "*PyTV*". 94% of particles have been identified (Fig. 8a). Without optimized parameters, the mean error is 5.3%. It depends on the understanding of our flow because with adapted filter parameters, this error can be reduced and reached values below 2%.

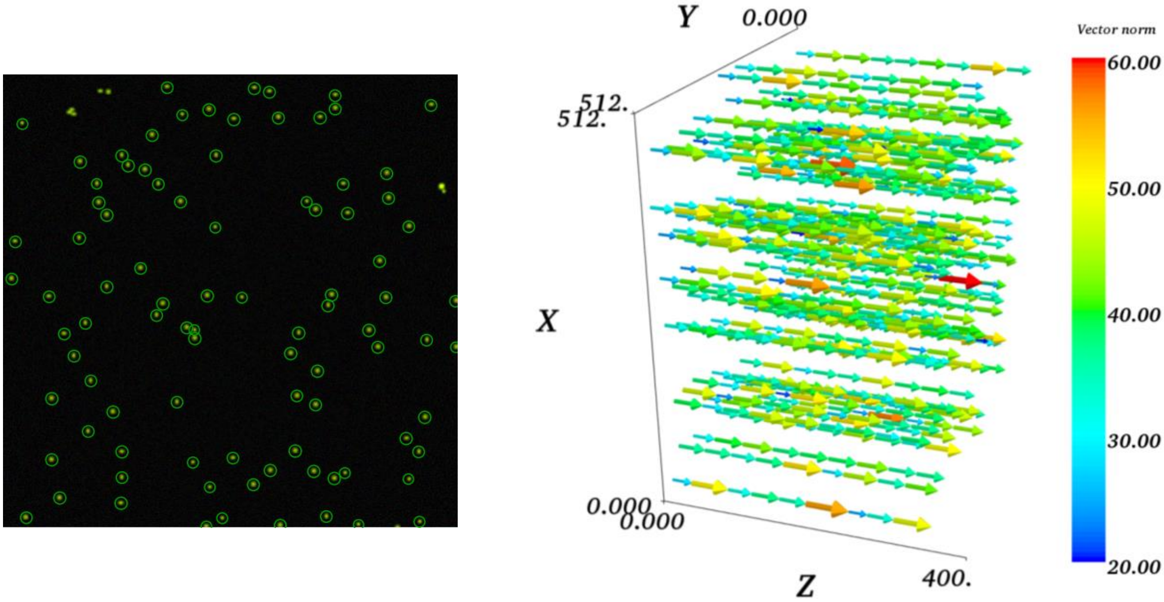

Fig. 8 (a) Identification result; (b) 3D vector field (True norm: 40)

We also used our program to estimate the global error on 3D particles' displacement by generating synthetic images whose particles move in each direction (x, y, z).

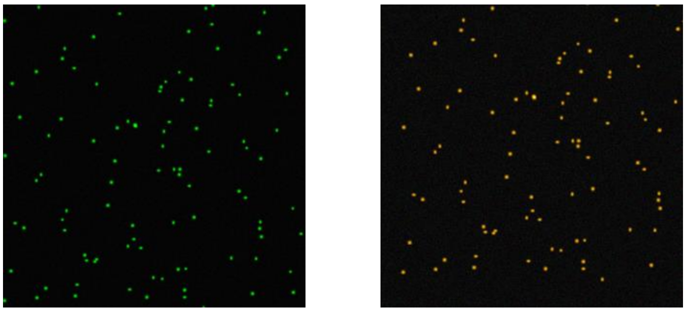

Fig. 9 Two synthetic images with displacement along x, y and z axis

The mean error is below 4% and vectors have a same direction (Fig. 10).

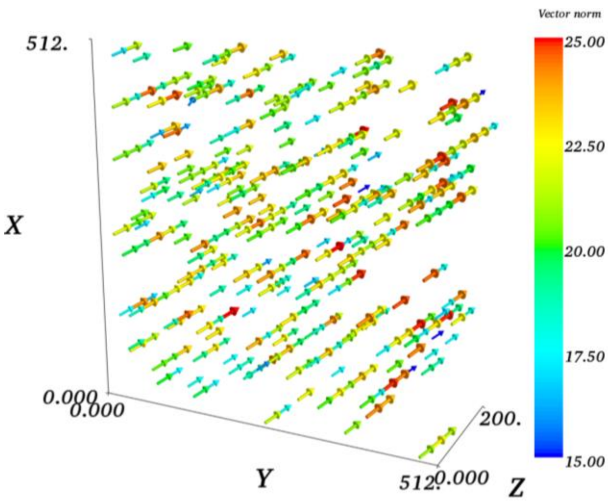

Fig. 10 3D vector field (True norm: 22.36)

Figure 11 gathers the displacement error along each axis.

Axis	True displacement	"PyTV" mean displacement	Error (%)
x	6	5.97	0.40
y	8	7.91	1.05
z	20	18.95	5.20

Fig.11 Error table

The most important error concerns the displacement along the z-axis. Our program is validated, but we can improve the color identification.

5 Conclusion

This experimental study concerns the development of a 3D particle tracking velocimetry method. We have adapted an experimental setup to work with 2D and 3D PTV. To process these images, we created a computer program using Python. We developed all the steps leading us to the determination of 2D and 3D velocity vector fields: pre-processing, identification of the particles, tracking and post-processing. For 3D program, we have implemented a calibration method to obtain a function giving us a position according to the color values of a particle. The colorimetric calibration will be improved and the 3D PTV method will be implemented on real experimental flows.
Thanks to this 3D particle tracking velocimetry method, our objectives are to characterize two-phase flows in the presence of one or more bubbles. One aspect of the study is fundamental and deals with the characterization of hydrodynamic instabilities (thermocapillary) growing at an interface (air / liquid or vapor / liquid).

References

[1] Prenel J P, Smigielski P (2000) Mécaniques des fluides – Visualisation, Techniques de l'ingénieur, AF 3 330, AF 3 331.

[2] Merzkirch W (1987) Flow visualization, Academic Press, New York.

[3] Bonnet D, Barthès M, Ramel D, Girardot L, Bailly Y, Guermeur F, Guyon D, Grenier A, Perrot Y, Ragot D, Vélocimétrie par PIV d'écoulements pulsés instationnaires sur banc moteur industriel, In: Proceedings of CMOI, Toulouse, France, 2010.

[4] Prenel J P, Bailly Y (2006) Recent evolutions of imagery in fluid mechanics: From standard tomographic visualization to 3D volumic velocimetry, *Optics and Lasers in Engineering*, vol. 44, pp 321–334. doi: 10.1016/j.optlaseng.2005.04.007.

[5] Fournier C (2003) Holographie numérique pour la vélocimétrie par images de particules : extraction des trois composantes du champ de vitesse dans un volume, PhD thesis, University of Saint-Etienne.

[6] Prenel J P, Bailly Y, Gbamele Y M (1999) Three dimensional PSV and trajectography by means of a continuous polychromatic spectrum illumination. In: Proceeding of the 2nd Pacific Symposium on Flow Visualization and Image Processing.

[7] Bailly Y, Gbamele Y M, Prenel J P (2000) Study of a three dimensional airflow by means of polychromatic spectrum illumination. In: Proceeding of the 9th International Symposium on Flow Visualization.

[8] Malfara R, Bailly Y, Cudel C, Prenel J P (2007) Evaluation of the Rainbow Volumic Velocimetry (RVV) process by synthetic images, *Journal of Flow Visualization and Image Processing*, vol. 14, pp. 1-15. doi: 10.1615/JFlowVisImageProc.v14.i1.10.

[9] Malfara R. (2006) Contribution au développement de la technique de Vélocimétrie Volumique Polychromatique - Application à l'analyse des Ecoulements 3D instationnaires en hydrodynamique. PhD thesis, University of Franche-Comté.

[10] Surhone L M, Tennoe M T (2010) Particle Tracking Velocimetry. Betascript publishing, UK.

[11] Keane R D, Adrian R J (1992) Theory of cross-correlation analysis of PIV images. *Applied scientific research*, vol. 49, pp. 191–215. doi: 10.1007/BF00384623.

[12] Bailly Y (2004) Méthodes optiques – actionneurs fluidiques – microsystèmes thermoacoustiques. Habilitation à diriger des recherches, University of Franche-Comté.

[13] Bruhat G. (1996) Cour de physique générale: Optique, Masson.

[14] Bonnet D. (2012) Vélocimétrie par images de particules (PIV) pour le diagnostic d'écoulements dans la boucle d'air automobile. PhD thesis, University of Franche-Comté.