



University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY
MASTER'S THESIS

Study program: Petroleum Technology/ Natural Gas Engineering	Spring semester, 2019 (Open access)
Author: Awais Ashraf	
Faculty supervisor: Prof. Rune Wiggo Time	
Title: Analysis of flame propagation using an imaging technique BOS (Background Oriented Schlieren) and PIV (Particle Imaging Velocimetry).	
Credits: 30 ECTS	
Keywords: <i>Flame analysis, flame speed, flame propagation, Background Oriented Schlieren (BOS), Particle Imaging Velocimetry (PIV).</i>	Number of pages: 94 + supplemental material/other: 4 Stavanger, 15.06.2019

Abstract

An experimental and simulation work has been conducted to study premixed flame propagation using imaging technique BOS (background oriented schlieren) and PIV (particle imaging velocimetry). Flame images obtained from experiment and computation using PIVlab 1.41 were used for studies.

The BOS (background oriented schlieren) technique is a newly established method for easily and inexpensive visualizing density gradient variation in compressible flows. The BOS setup design consists primarily of a background with random dots and a camera targeted on the dots. Variation of the density between the uninterrupted image and the distorted image resulted in a dots shift that is used by the optical flow method to generate displacement field.

We present a simple 2D imaging method for measuring this distortion and then show how it can be used to visualize gas and liquid flows. In addition, several research questions associated to combustion and fluid mechanics are explored using high speed direct/schlieren imaging, PIV techniques, etc.

Acknowledgement

I would like to pay my gratitude to Prof. Rune Wiggo Time for his consistent support and motivation throughout the achievement of this thesis. His remarkable and kind benefaction through word and perception has been of countless worth.

In addition, I would like to thank Senior Engineer Hermonja A. Rabenjafimanantsoa, Benja, for providing me opportunity to work in laboratory and handed over keys to work in my desire schedule.

Finally, I want to acknowledge the support, and love of my family – my parents and siblings. They all kept me going and this work was not possible without their support.

Author

Awais Ashraf

Contents

Abstract	ii
Acknowledgement	iii
List of figures.....	vi
List of tables.....	viii
Abbreviations.....	ix
Chapter 1: Introduction	1
1.1. Project background	1
1.2. Motivation.....	1
1.3. Project Goal	2
1.4. Arrangement of the thesis	2
Chapter 2: Literature Review	3
2.1. Fuels and the combustion process.....	3
2.1.1. Combustion	3
2.1.2. Premixed flame	4
2.1.3. Non-premixed/diffusion flames	5
2.1.4. Transition from Laminar to turbulent flames.....	6
2.1.5. The buoyant plume.....	7
2.2. Optical experimental techniques in basic research on combustion.....	9
2.2.1. Background Oriented Schlieren and similar techniques	10
2.2.2. Optical principle of BOS.....	13
2.2.3. The advantages of BOS.....	14
2.2.4. The disadvantages of BOS	14
2.3. What is an Image?.....	15
2.3.1. Image Analysis problem?.....	16
2.3.2. Image Storage and Compression.....	17
2.3.3. Measurements	18
2.4. What is noise?	19
2.4.1. Filter	19
2.5. Particle Image Velocimetry (PIV)	20

2.5.1.	Implementation and architecture.....	21
2.5.2.	Image pre-processing	22
2.5.3.	Calibration.....	23
2.5.4.	Computation of the cross-correlation function	24
2.6.	Divergence	24
Chapter 3: Experimental Setup		25
3.1.	Experiment and required resources.....	25
3.1.1.	Video capturing.....	25
3.1.2.	Video split program/ software.....	26
3.1.3.	Data processing/BOS image processing software.....	27
3.1.4.	Results evaluation	28
3.2.	Challenges:.....	29
Chapter 4: Results & Discussion		30
Chapter 5: Conclusion & Future Work		80
References.....		82
Appendix A.....		86
Appendix B		94
Appendix C.....		104
Appendix D.....		106

List of figures

Figure 1: Premixed burner schematic illustration [4]	5
Figure 2: Premixed and Non-premixed flame [5]	6
Figure 3: Flame height vs jet velocity, showing flow transition to turbulence [6]	7
Figure 4: The buoyant plume development [2]	8
Figure 5: Diagram of axisymmetric plume [2]	8
Figure 6: Digital image processing and its bionics [39]	9
Figure 7: Development of measurement techniques [23]	11
Figure 8: Background Oriented Schlieren	12
Figure 9: Actual image area with corresponding magnified view to show pixel intensities [24]	16
Figure 10: Image analysis process steps [24]	17
Figure 11: Filter processing for de-noising [40]	19
Figure 12: PIV principle [28]	20
Figure 13: Workflow of PIV analysis in PIVlab [28]	21
Figure 14: Noise reduction process [40]	22
Figure 15: The consequence of pre-processing techniques [28]	23
Figure 16: Experiment flow diagram	25
Figure 17: Figure describes inputs (left side) and output of MATLAB program	28
Figure 18: Artificial experiment for 0.01 pixel shift produced by using MATLAB program	30
Figure 19: Artificial experiment for 0.05 pixel shift	31
Figure 20: Input images of the experiment using white paper	32
Figure 21: PIV analysis images of experiment on white paper	33
Figure 22: Schematic diagram of BOS experiment	34
Figure 23: PIV analysis images 1 at reference time for 30 fps	35
Figure 24: PIV analysis images 2 at reference time for 30 fps	36
Figure 25: PIV analysis images 3 at reference time for 30 fps	37
Figure 26: PIV analysis images 4 at reference time for 30 fps	38
Figure 27: PIV analysis images after 5 sec for 30 fps	40
Figure 28: PIV analysis images after 10 sec for 30 fps	42
Figure 29: PIV analysis images 1 at reference time for 60 fps	44
Figure 30: PIV analysis images 2 at reference time for 60 fps	45
Figure 31: PIV analysis images 3 at reference time for 60 fps	46
Figure 32: PIV analysis images 4 at reference time for 60 fps	47
Figure 33: PIV analysis images after 5 sec for 60 fps	49
Figure 34: PIV analysis images after 10 sec for 60 fps	51
Figure 35: PIV analysis images 1 at reference time for 240 fps	53
Figure 36: PIV analysis images 2 at reference time for 240 fps	54
Figure 37: PIV analysis images 3 at reference time for 240 fps	55
Figure 38: PIV analysis images 4 at reference time for 240 fps	56
Figure 39: PIV analysis images after 5 sec for 240 fps	58
Figure 40: PIV analysis images after 10 sec for 240 fps	60
Figure 41: Effect of surrounding air and flame plume at stable position	61
Figure 42: Effect of rolling shutter on moving propeller [37]	62

Figure 43: Diagram depicting experimental setup using mirror	63
Figure 44: PIV analysis images 1 using mirror at reference time for 30 fps	64
Figure 45: PIV analysis images 2 using mirror at reference time for 30 fps	65
Figure 46: PIV analysis images 3 using mirror at reference time for 30 fps	66
Figure 47: PIV analysis images using mirror after 5 sec for 30 fps.....	68
Figure 48: PIV analysis images using mirror after 10 sec for 30 fps.....	70
Figure 49: PIV analysis images 1 using mirror at reference time for 60 fps	71
Figure 50: PIV analysis images 2 using mirror at reference time for 60 fps	72
Figure 51: PIV analysis images 3 using mirror at reference time for 60 fps	73
Figure 52: PIV analysis images 4 using mirror at reference time for 60 fps	74
Figure 53: PIV analysis images using mirror after 5 sec for 60 fps.....	76
Figure 54: PIV analysis images using mirror after 10 sec for 60 fps.....	78
Figure 55: Internal reflection in mirror [38]	79

List of tables

Table 1: Types and examples of combustion [3]	4
Table 2: Comparison between number of dots and diameter for background pattern	26

Abbreviations

BMP	Bitmap
BOS	Background Oriented Schlieren
CCD	Charged Coupled Device
CFD	Computational Fluid Dynamic
CLAHE	Contrast Limited Adaptive Histogram Equalization
CMOS	Complementary Metal-Oxide Semiconductor
DCC	Direct Cross Correlation
FFT	Fast-Fourier Transform
FPS	Frame per Second
GUI	Graphical User Interface
ICE	Internal Combustion Engine
JPEG	Joint Photography Experts Group
PDFA	Phase Detection Autofocus
PIV	Particle Imaging Velocimetry
PLIF	Planar Laser Induced Fluorescence
PNG	Portable Network Graphics
Re	Reynolds number
TIFF	Tagged Image File Format

Chapter 1: Introduction

1.1. Project background

Since ancient times, human beings are using flame and fire to fulfill the different daily purposes. Today this process of burning is still being extensively used in industrial applications. Approximately 90% of the usage of this energy is linked to combustion like processes, for instance gas turbine engine and ICE (internal combustion engine) etc. Researchers have developed many models and methods to comprehend the behavioral mechanism of this complicated phenomenon. Combustion plays an important role in modern industry. Understanding of the combustion process is a quickly emerging area and it involves a broad variety of subjects like flame dynamics, ignition, flame quenching and the mechanism of chemical reaction.

In experimental fluid dynamics, generally three approaches are being used to visualize the flow: visualization of surface flow, particle tracer techniques, and optical methods. Surface flow visualization discloses the flow streamlines as the solid surface methodologies. One example of this technique is colored oil applied to the surface of a wind tunnel model. Particles like smoke can be added to a flow to trace flow movement. A part of the complicated flow pattern could be illuminated by laser sheet to understand this complex phenomenon. By Assuming that the particles track the streamlines of the flow closely, we can not only visualize the flow but also measure its velocity using a method known as PIV (particle image velocimetry). Finally, by changing their optical refractive index, some flows reveal their patterns. These techniques help to visualize the phenomenon using optical techniques generally known as the schlieren, shadowgraph, interferometry and more recently, the BOS (background oriented schlieren) technique.

1.2. Motivation

Optical diagnosis plays an important role among all combustion diagnostic techniques as it appears to have certain advantages like it is intuitive, non-intrusive and offering large information sets contrast to single point measurement. In recent times, for both quantitatively and qualitatively visualization of combustion processes, optical techniques such as

shadowgraph/schlieren, PIV and PLIF, have been extensively used. The understanding of combustion characteristics can be increased by combining optical visualization, high-speed imaging and image processing. Most optical techniques are based on 2D (two-dimensional) and it is fairly new concept to measure flame dynamics in 3D (three-dimension) space.

The objective of this thesis is to develop the BOS visualization method for flame analysis using a simple optical setup. Another objective of this study is to achieve natural understandings into the phenomena of combustion/fluids flows using a mixture of optical approaches.

1.3. Project Goal

The purpose of this project was to develop a background oriented schlieren technique for flame propagation analysis. A BOS method will allow not only capturing qualitative images of compressible flow features but also extracting quantitative information about the flow.

1.4. Arrangement of the thesis

This thesis has been organized according to guideline provided by the University of Stavanger. The dissertation has been broke up into five chapters. Chapter 1 highlights goal and motivation. Chapter 2 presents theories related to Combustion, flame, BOS (background oriented schlieren), Image, Image Analysis and PIV (particle imaging velocimetry). Chapter 3 represents experimental setup. In chapter 4, I have represented PIV results and discussion. In the end, chapter 5 presents conclusion of the thesis.

Chapter 2: Literature Review

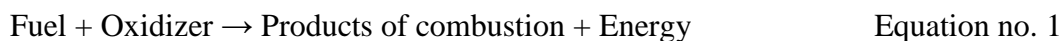
Combustion involves chemical reaction of air (oxygen) and any combustible material (solid, liquid or gas), which results fire. It offers heat and power for our domestic and industrial use, but it could have serious effects on the human health and environment. In 1993, United Kingdom suffers around 800 human lives and around £1000 million [1]. But over the past two decades, there is no significant increase in losses due to enhancing technical competence of fire service and smart fire defense systems.

2.1. Fuels and the combustion process

Fuel can be anything which is burning regardless of state of matter like methane, petrol or a wooden item in room. Under proper condition, all will burn producing combustion products and discharging heat.

2.1.1. Combustion

“Combustion is a complex sequence of exothermic chemical reaction between a fuel and an oxidant, during which heat and light are emitted to form a flame”. A flame, which is seen as a visible portion, is a self-reliant circulation of a localized combustion zone. In a typical chemical reaction, for example:



A comprehensive overview of combustion types, flow types and their examples has been summarized in **Table 1**. From the Table 1, it can be seen that combustion system can be classified on the basis of preparation of the fuel (or oxidizer mixture). There are two major types of mixing; premixed and non-premixed. Their flow could be laminar or turbulent. Bunsen and flat flame are examples of laminar flow, while gas turbine and spark ignition are examples of turbulent flow [3].

Table 1: Types and examples of combustion [3]

Type of mixing	Flow type	Example
Premixed	Laminar	- Bunsen flame - Flat flame
	Turbulent	- Gas turbine - Spark-ignited engine
Non-premixed	Laminar	- Wood fire - Candle
	Turbulent	- Diesel engine - Pulverized coal combustion

2.1.2. Premixed flame

Premixed flame is a combustion reaction that occurs before chemistry starts when fuel and oxidizer are mixed. The flame front movement is a fluid convection and flame propagation superposition. A simplified example of premixed flame is a Bunsen burner is illustrated in **Figure 1**. Chemical process dominates the flame speed of premixed flame for laminar flow of reactants. Hence, a stationary flat flame front will form, if the flowrate of reactant is equal to the flame speed. If the flow rate exceeds the velocity of flame, a conical shape will be displayed in the flame. If the flow rate is lower than the speed of flame, the front moves upstream until the blend exceeds its flammability boundaries or the flame is quenched owing to flow conditions modifications. One of main hazard with premixed flame is flame flashback which eventually leads to explosion [4]. As the speed of flame propagation increases, flame tends to move from laminar to turbulent flame which increases mass and heat transfer.

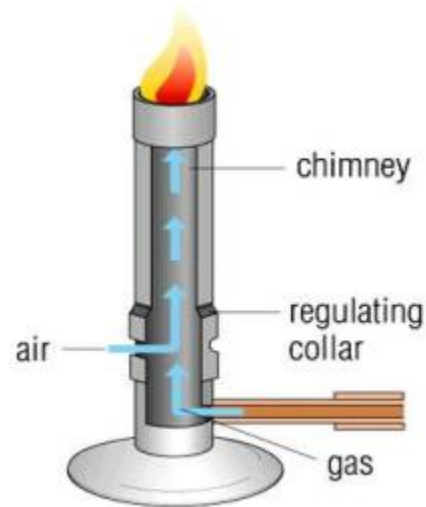


Figure 1: Premixed burner schematic illustration [4]

2.1.3. Non-premixed/diffusion flames

Diffusion flame's main characteristic is that fuel and oxidizer are originally distinct and combustion happens in the mixing area. A candle or Bunsen flame with the air inlet port closed is an instance of diffusion flame. The appearance of the flame will hinge on fuel's nature and the speed of fuel jet in relation to the surrounding atmosphere [4].

The blending and combustion of fuel and oxidizer takes place simultaneously in non-premixed flames. At the fuel and air interface, the diffusion flame front is formed. Diffusion flames are generally mixed-controlled or diffusion-controlled, rather than in premixed flames kinetically-controlled [3].

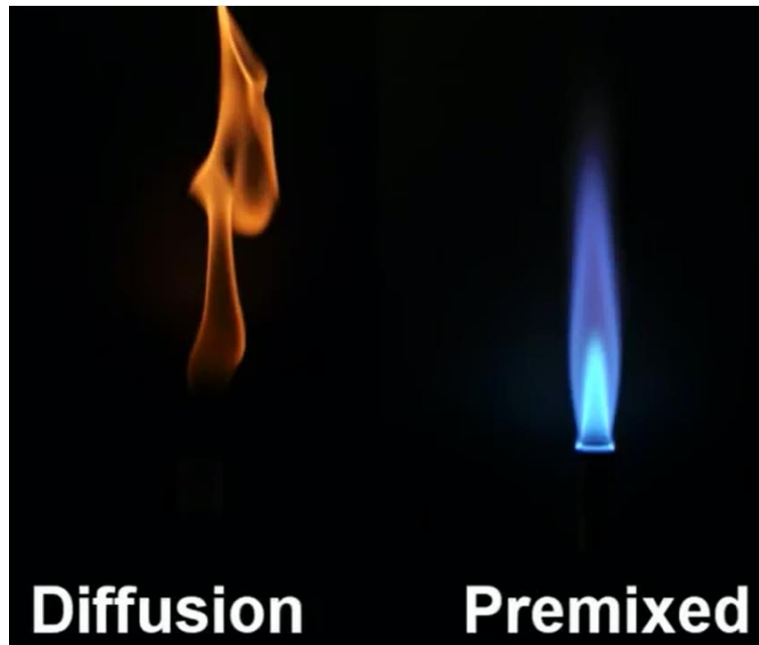


Figure 2: Premixed and Non-premixed flame [5]

2.1.4. Transition from Laminar to turbulent flames

Combustion could occur in laminar or a turbulent flow environment, but it can also take place in laminar-to-turbulent transitional flow condition. Laminar flow occurs when fluid flows without lateral mixing in parallel layers between the layers. But turbulent flow is categorized as chaotic, where adjacent layers constantly upset each other. Viscous force dominates the laminar flow (low Re), while inertial force controlled the turbulent flow (high Re). The flow is partially laminar and partially turbulent in some circumstances and transition takes from laminar to turbulent flow in this condition [7].

The behavior of the flame could be influenced by two parameters, flow conditions and buoyance forces induced by chemical reaction process. With the increase in Re , it is possible to observe a shift from laminar to turbulent flow. The flow completely changes to turbulent with the increase in Reynolds number. With the more increase of Re , a flame which is detached from nozzle can be established. At a critically high value of Re , the flame can be finally blown-off [7].

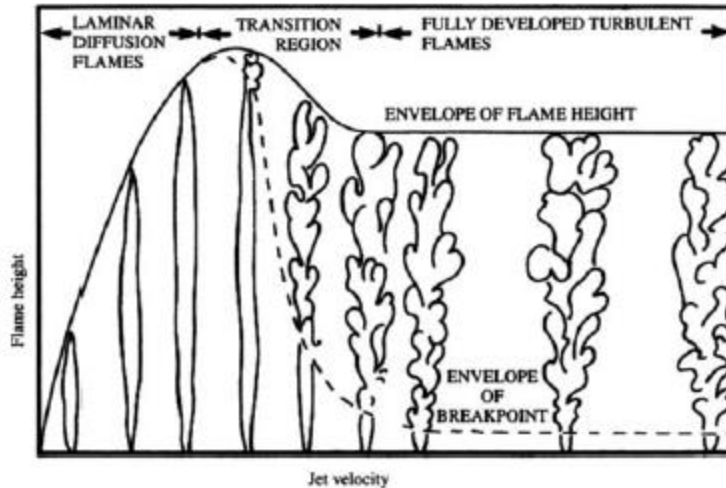


Figure 3: Flame height vs jet velocity, showing flow transition to turbulence [6]

When jet velocity increases, turbulence starts from top of flame and extends to bottom nozzle but will never touch it (figure 3). In addition, the influence of air rises through eddy mixing, which decreases flame height from laminar section to constant value in fully turbulent system.

2.1.5. The buoyant plume

The notion of buoyancy always presented in association to natural convection. If there is a distinction in densities between liquids, then the buoyancy force will raise the fluid with less density compared to its atmosphere. To define this convective column, rise from a source of heat, the term ‘buoyant plume’ is used. Plume contour is defined by its interaction with the neighboring fluid (mostly air) (figure 4) [7].

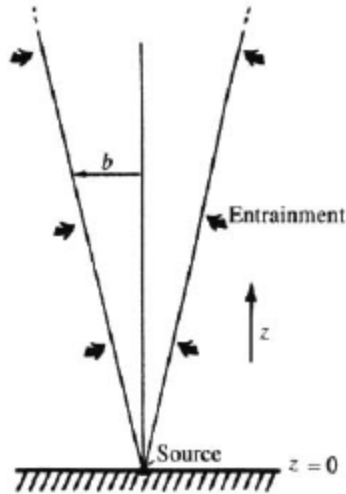


Figure 4: The buoyant plume development [2]

The ideal plume is axisymmetric, extending vertically to a level where force of buoyancy becomes too feeble to overcome viscous drag (figure 5). A temperature inversion can form under specific atmospheric environments that will cause rising smoke plume to spread laterally (i.e. interaction with ceiling) [2].

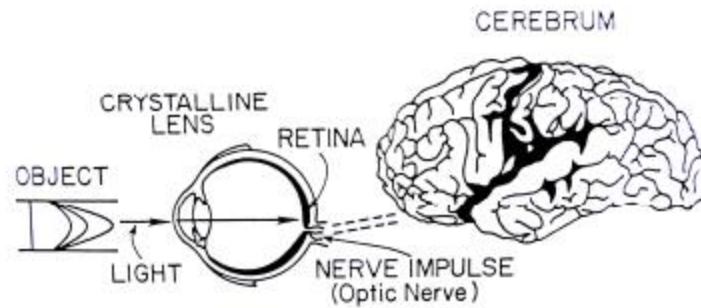
In the flame plume, the vertical movement of buoyant gasses causes entraining of the surrounding air. This not only provides oxygen/air for fuel to be combusted, but it also cools and dilutes fire products when they grow above the flame and cause a constant rise in the size of smoke produced by the fire.



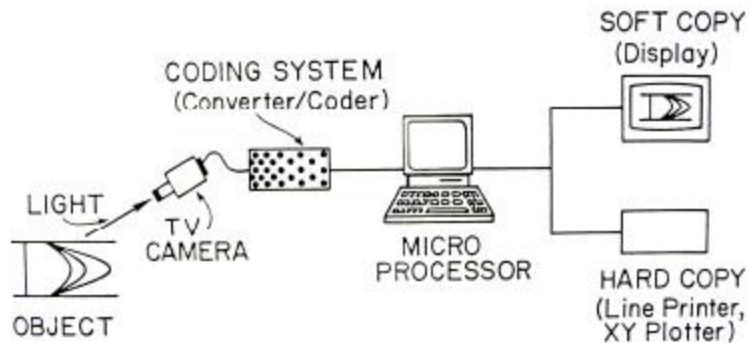
Figure 5: Diagram of axisymmetric plume [2]

2.2. Optical experimental techniques in basic research on combustion

Leonardo da Vinci (1452-1519) identified observational science in experiments during Renaissance era. He was the first person to explain flow visualization phenomena, such as eddies formation and wave propagation, etc. [7]. Different elements are launched into the fields as a tracer or contrast medium in traditional flow visualization techniques to discriminate the object to be observed. To achieve high contrast and resolution in images, flow field needs to be illuminated. Recently, the flow visualization approaches have advanced to a new era with the speedy progress of computational methods and computer color graphics. **Figure 6** illustrates the bionics and digital image processing procedure.



(a) Bionics image processing by eye and brain



(b) Digital image processing

Figure 6: Digital image processing and its bionics [39]

Figure 6, (a) represents the identification of an object with human eye. In order to focus on the retina, light rays reflected from the object while passing through the lens. Light rays in the retina are stimulating millions of optical receptors. Figure 6, (b) shows the digitalization of image and digital image processing process. For an object to be automatically identified, a system similar to human vision is needed. A camera is analogous to the human eye and captures the image of object. A coding system converts the camera image into digital image, a microprocessor performs the processing function. The output could be softcopy or a hardcopy (printed version) [7].

The advancement in the optical investigative techniques has played crucial role in flow visualization, pressure and distribution of temperature, velocity measurements, etc. The results of visualization help to understand physics of the complex phenomena of flows, encourage theory articulation and also offer significant feedback for engineering designs.

Optical diagnostic techniques have been applied in combustion research from decades. The invisible flow pattern significantly affects the ignition and combustion in combustion phenomena in addition to visible light emission. Combustion includes chemical reactions and excessive temperature, non-intrusive optical diagnostics therefore demonstrate the huge benefits of obtaining visualization and measurement. One of the influential flow visualization tools is schlieren and shadowgraph technique that can visualize invisible variations in density and has been extensively used to study combustion, fluid mechanics and aerodynamics, etc. Particle imaging velocimetry is used by introducing tiny particle into flow field as a standard technique for measuring velocity. PLIF (planer laser induced fluorescence) can tackle specific species distribution by recognizing the appropriate light emission. Most diagnoses are two-dimensional. In diagnostic techniques such as stereo imaging and tomography, great efforts have been made to obtain information in three dimensions. In exploring and developing three dimensional diagnostics, great efforts are needed [7].

2.2.1. Background Oriented Schlieren and similar techniques

The BOS technique is comparable to observing the effect of local density gradients like heat haze, mirage or fata morgana, where the observation of the observer distorts a distant object or

even mirror the perceived image **Figure 7** (a). Another effect, known as shadow effect (**Figure 7** (b)), is observable with the naked eye but with optical instrumental aid it became more evident. The density gradient between the source of light and the surface with monotonous brightness and color generates this effect. The changes in intensity ΔI at every point on the surface are directly proportional to the second air density derivative. The shadowgraph is known as the picture of these varying light intensities [23].

Techniques for measuring optical density such as shadowgraphy, schlieren or interferometry are well established and have been in use for many centuries. They are based primarily on variation in the refractive index and allow flow density variations to be observed [8].

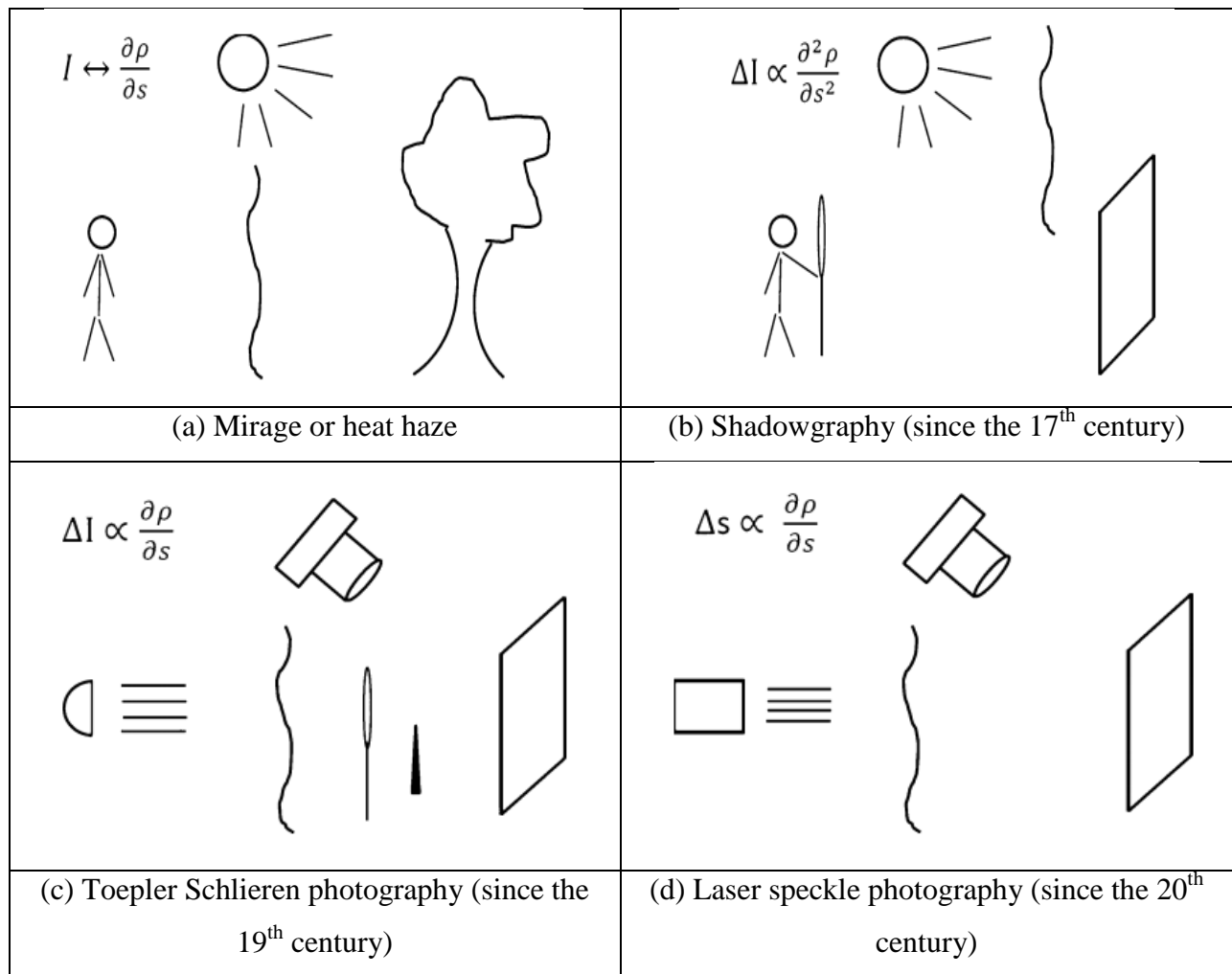


Figure 7: Development of measurement techniques [23]

The study of the density gradients became important in the field of fluid dynamics when aerodynamics developed transonic. An important driver for enhancement and application of observation of fluid dynamics was the experiment of bullets shooting and the study of flow fields in the work of Ernst Mach. This method is most frequently used for the visualization of fluid density in Toepler's schlieren photographic work (**Figure 7, (c)**). Typically, its experimental configuration comprises of spherical mirrors and an aperture. The light intensity variation ΔI captured here are proportional to the first fluid density derivative.

Laser speckle interferometry is another method that is very close to BOS method (**Figure 7, (d)**). The screen was illuminated with laser light by the density gradient under investigation, which produces a speckled interference pattern. The generated and optically evaluated displacement Δs is proportional to the first density derivative. The schlieren technique is typically centered on the valuation of image displacements. Incoherent light can illuminate the background and imaged it through a fluid. Illustration of BOS is presented in **Figure 8**.

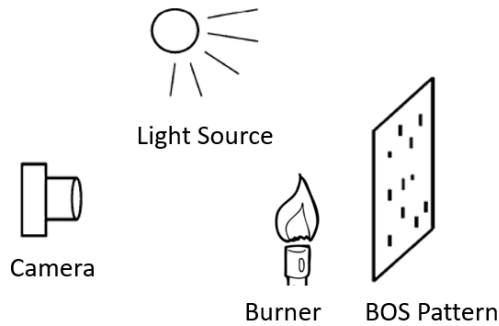


Figure 8: Background Oriented Schlieren

The experimental arrangement for typical arrangements of the BOS experiment can be organized by putting the background target with the randomly generated dots on one side and a camera on the opposite side [10, 11]. The background target is important in image capturing and can affect the spatial resolution [11, 12]. Many factors can influence the measurement and most important factors are the camera's aperture, focal length, distances between the optical elements and the size of the interrogation window [11, 13, 15, and 16].

Digital valuation and white-light illumination are also well-known in speckle interferometry, So, BOS could also be known as “white-light speckle density photography”. However, the most common name for schlieren method is “background oriented” that actually explains how the camera focuses on objects behind the investigated flow [9].

2.2.2. Optical principle of BOS method

In the present research, the basic idea of this optical technique is based on the variation of air refractive index due to density gradients. The Gladstone-Dale equation explains the relationship between density variations and refractive index:

$$\frac{n-1}{\rho} = G(\lambda) \quad \text{Equation no. 2}$$

Where n: refractive index, ρ : density (kg/m^3) and $G(\lambda)$: Gladstone-Dale number (m^3/kg).

The method is best compared to density speckle photography as explained by Debrus et al. (1972), Kopf (1972) and Wernekinck and Merzkirch (1987) in an updates edition. The expanded laser beam and a ground glass generates a pattern, a laser spackle pattern is replaced by a background pattern with random dots. The laser pattern should have the high spatial frequency that could be helpful to have images with reasonable distinction.

In the first step, a reference image is generated by recording the background pattern detected through air at rest. Additional exposure through the flow under analysis results in locally displaced background pattern image in the second step. Then both resulting images by the exposures can examine by methods of image correlation. Existing assessment algorithms, for instance for particle image velocimetry, which have been established and optimized, can then be used to determine pattern displacement at various positions.

The main distinction between BOS method and the other techniques described above is that no optical devices are needed to produce background. The only optical tool required is a lens on the camera. The camera’s main focus is the background pattern that produces an image that is moderately analogous to a particle image and that’s why the technique is known as “background oriented” [8].

In different studies, similar methods of observing and comparing background pattern were observed [12, 17]. If the spatial resolution of the background does not matter, the only main requirement for the target is a high pattern contrast [12] that can be readily accomplished by printing dots on a paper with a standard printer. The literature also recorded the effective use of natural background patterns [10, 11, and 18]. BOS is able to measure two in-plane, first-order gradients parallel to the background target and the image sensor, analogous to the schlieren method, which are integrated along the line of sight [13, 14, 16, and 19].

2.2.3. The advantages of BOS

It not only provides the quantitative measurement, but it also has some other benefits and can be summarized as: simpler modification compared to traditional schlieren methods, reduced number of optical elements, no restriction on measurement object size and straightforward computation [11, 13, and 14]. The potential of using natural backgrounds to image is an advantage of BOS [18, 21]. In addition to temperature or density measurement, the mixture of PIV and BOS can provide valuable data regarding velocity circulation in a flow.

2.2.4. The disadvantages of BOS

(a) Limited resolution:

BOS always delivers less image resolution than comparable traditional schlieren methods because image cross-correlation algorithms average their interrogation window size [23]. Similarly, an increase in BOS sensitivity requires a larger interrogation window, resulting lower resolution [13]. In discrete flow features such as shock waves, resolution loss is most noticeable.

(b) Difficulty in attaining sharp focus on both subject and background:

It is clear that camera sensor needs to have a sharp focus on the BOS background [23], but what about schlieren object (in my case flame) under study? The camera's depth-of-field can theoretically extend to include both, but this can lead to an impractically large distance from camera to schlieren object, especially in a laboratory where dimensions are restricted. In order to

minimize this problem, Meier [11] recommends a small camera aperture and a short distance to the background from the schlieren object, although the latter also reduces BOS sensitivity.

(c) Non-parallel illumination:

BOS shares the problem of having a small camera lens looking at a large background grid along non-parallel rays with direct shadowgraphy and lens-and-grid schlieren. This complicates the quantitative measurement of schlieren object. Avoiding these pitfalls is an important reason for expensive lenses and mirrors in schlieren instruments.

(d) Vibration sensitivity:

BOS compares a flow-on image to a previously acquired on-flow image. Vibration, or anything other than schlieren object that changes between these two exposures, is a problem [13].

BOS observations are extremely sensitive in the experimental instruments to any changes and deflection. Any disturbance or vibration on the optical path may result in elevated evident background pattern displacement and trigger BOS signal deformation. Precautions should therefore be taken to eradicate any vibration in the optical path.

2.3. What is an Image?

An image is a three-dimensional array of numbers in its simplest form that shows intensities and spatial coordinates (x and y- axis) of pictured item (figure 9). To improve or make numerical measurement of attributes in an image, mathematical calculations are carried out on numbers of an array. In the digital world, pixels are typically small square picture element, by which an image is composed. The number of light photons that strike a camera's detector determines each pixel's intensity or gray level. Typically, images ranges from 256*256 pixels to 4096*4096 pixels using specialized imaging accessories. Resolutions and sensitivities of cameras are getting better day by day.

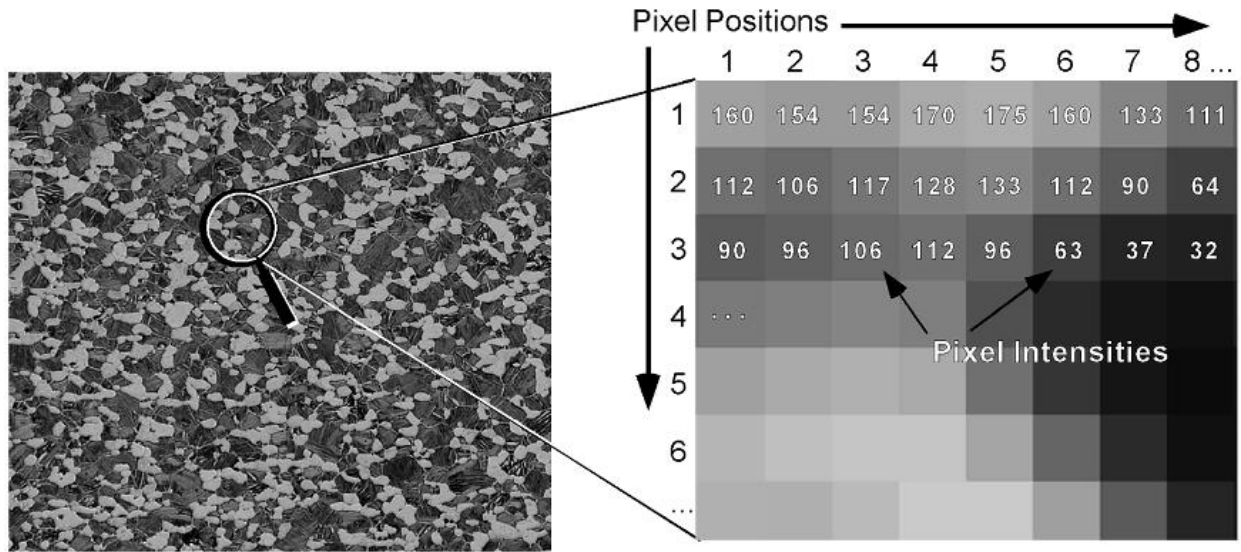


Figure 9: Actual image area with corresponding magnified view to show pixel intensities [24]

2.3.1. Image Analysis problem?

The process of analyzing the visualized scene incorporates specific steps that lead to an improved image or data that can be used for further interpretation. A conclusion is essential at each step to be able to accomplish the next step, as shown in figure 10. Moreover, different algorithms can be used for every step to accomplish the anticipated effects and/or measurements.

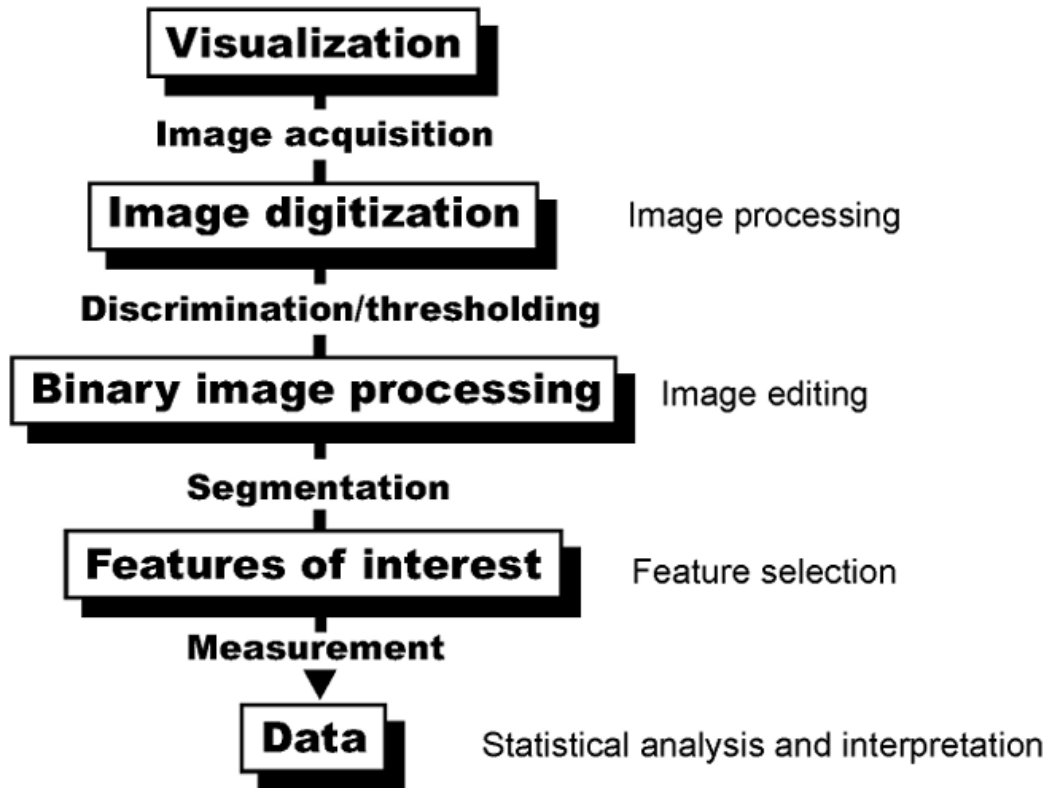


Figure 10: Image analysis process steps [24]

2.3.2. Image Storage and Compression

Many systems save visual data onto a permanent medium (e.g., floppy, hard, and optical disks) using exclusive algorithms that compress the images to some extent. There are standardized compression algorithms, like, JPEG (joint photography experts group) and TIFF (tagged image file format). These algorithms make it heavy to transport the images, but several systems allow exporting images into standard formats. When the images are stored in standard formats, considerable loss of information could occur because of compression of images, so, great care should be exercised when storing images.

Therefore, it is generally recommended that the images for analyzing purposes should be stored with very limited compression (no less than half the original size) or no compression as compared to the images that are for archival and visualization purposes only that could be compressed as desired. The information loss connected with compression of images depends on

the type and size of image being compressed and gray-level range of the structures to be quantified.

There are four commonly used image formats, JPEG, TIFF, BMP, and PNG. JPEG is commonly used in domestic devices. Images can be very large these days with good quality cameras around. When image is compressed to different formats, following is the sequence according to size:

BMP > Tiff > PNG > JPEG

- JPEG achieves very good compression ratios, but is lossy. Take an image, put it through JPEG compression process, you'll get a very small image, but it will be impossible to recover from that image the one that you started with.
- TIFF, the tanked image file format, is generally, though not always, lossless. TIFF has many settings and options, and it's possible to create a lossy version of TIFF very, very easily. So TIFF should be used with care.
- PNG is a lossless format. So there is no danger with a PNG images that you will destroy an image in the compression process.
- BMP is perhaps the simplest of the available formats. It's lossless so it's safe to use, but doesn't allow you to add any metadata or any layers. In most scientific applications, BMP or PNG are the safest to use.

2.3.3. Measurements

The essence of image analysis is making a measurement or series of measurements that quantify some aspect of the image. Image analysis measurements can be made manually, automatically, or even by comparing a series of images that already have been measured. Manual measurement usually involves counting features, such as points, intersections, and intercepts, but it also involves measuring length. Area and volume, on the other hand, usually are derived values. All of these measurements are easily calculated by the computer for every feature in the image(s).

2.4. What is noise?

Whenever we're taking images for use in scientific experiments, we're faced with a number of problems. One of these is the camera or the microscope or whatever device it is that we're using to take the images doesn't actually capture exactly what's out there in the real world. Whatever we're seeing in the image has been distorted ever so slightly by the way that we're capturing it and the device that we're using and the process by which that device turns the lights on the signal into some kind of digital data. This destructive process can be called image noise, where we're distorting that signal that we're recording [40].

To measure and estimation of the noise level appears to be an impossible task from a single image: it should be acknowledged that whether texture, color or lighting from image itself is the cause of the image variations, or whether it is due to noise. Real CCD camera noise, unfortunately, depends heavily on the level of image intensity [27].

2.4.1. Filter

Noise reduction processes usually take the form of filters. These work by examining a small window of pixels called a kernel, for example 3 x 3 pixels in size, then recalculation the middle pixel based on the neighboring pixels in this window. This window is then moved across the image typically a pixel at a time and applied again until the whole image has been recalculated. We can perform different calculations on the pixels in the kernel to handle different kinds of noise, but in all cases, we are using local image statistics (such as a mean) to recalculate the center value based on the context of its neighbor's [40].

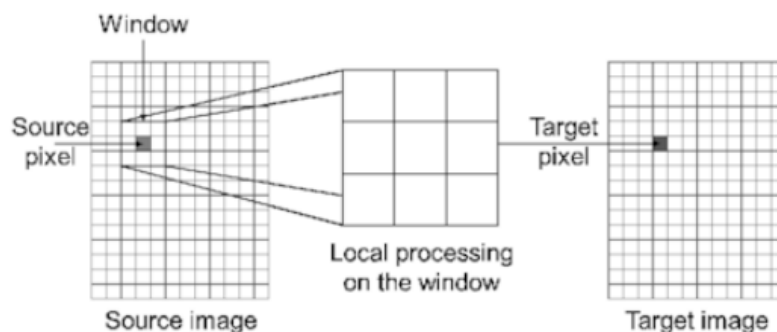


Figure 11: Filter processing for de-noising [40]

2.5. Particle Image Velocimetry (PIV)

PIV (Particle image velocimetry) is a widely accepted measurement technique for studying turbulent flows structure since it offers to measure quantitative data on the velocity field's spatial structure.

Particle image velocimetry (PIV) is an accepted approach for visualization of non-intrusive, qualitative and quantitative flow. Here, for PIV analysis in MATLAB, a GUI-based open-source tool (PIVlab) was used. Multiple embedded MATLAB features are used and subsequent computing is facilitated by providing a close connection to the common MATLAB [28].

An illustration of PIV principle has been shown in **Figure 12**. In PIV, fluid movement (either liquid or gaseous) is observed by enlightening a small sheet of fluid having neutrally buoyant and reflective footprint particles. Parallel sheet position is given for digital image sensors to capturing particle movements. Two images (A and B) of the sheet are captured at t_0 and $t_0 + \Delta t$ for PIV analysis. From distance and difference in time (Δt) between two images, velocities can be derived from the illuminating sheet. The particle displacement is determined by assessing the cross-correlation of many small sub-images (interrogation areas). The correlation provides the most likely particle shift between image A and image B moving on a straight line [29].

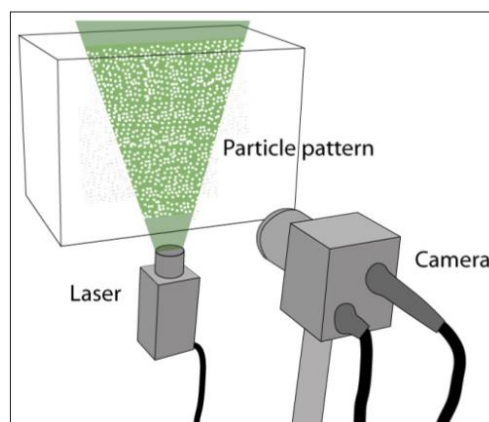


Figure 12: PIV principle [28]

2.5.1. Implementation and architecture

MATLAB has a lot of program applications and PIVlab is one of the image processing applications. PIV analysis involves three main steps (pre-processing, image evaluation and post-processing, figure 13).

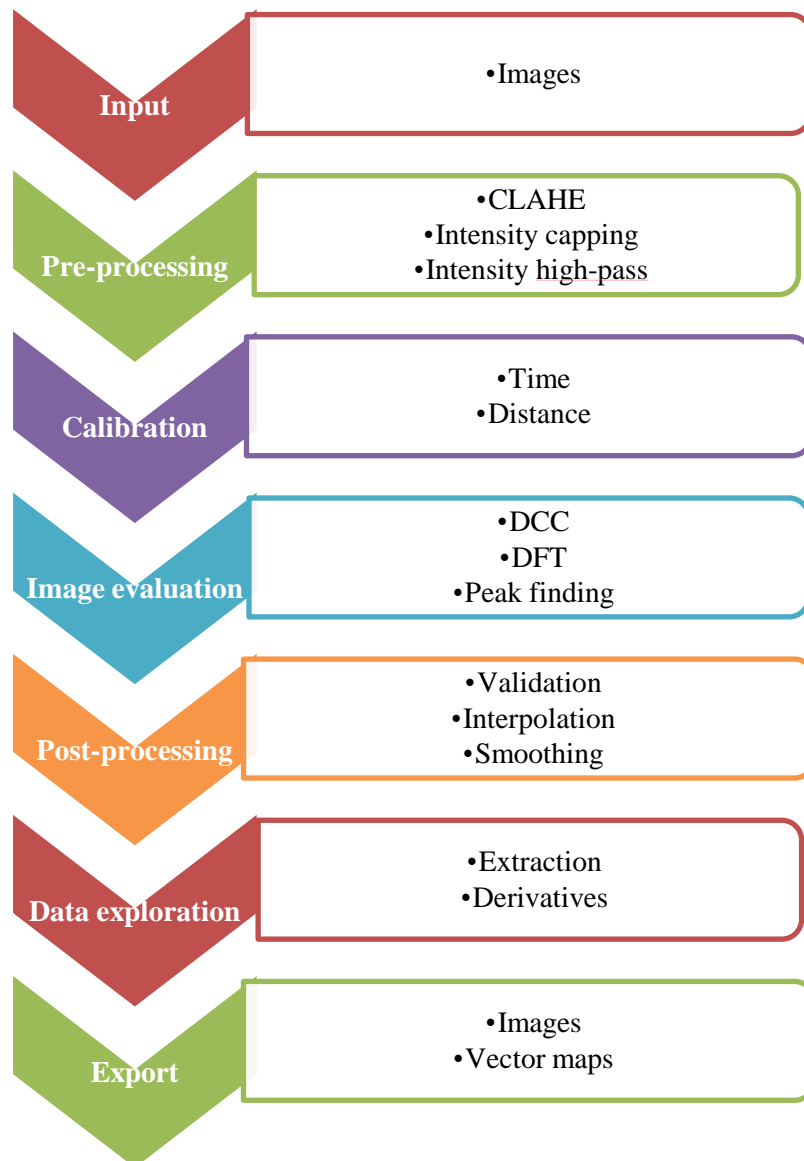


Figure 13: Workflow of PIV analysis in PIVlab [28]

GUI based interface of PIVlab has all of these steps. Workflow is menu-based, starts with the image input and continues to the bottom of the figure.

An overview of relevant technique and features will be provided in the following section:

2.5.2. Image pre-processing

A general way for better measurement is to improve the images before the actual image correlations used [29, 30]. Some of the pre-processing techniques are given below: (figure 15)

(a) Histogram equalization:

To enhance the readability of the image data in medical imaging, Contrast limited adaptive histogram equalization (CLAHE) was established [31]. CLAHE works on small sections of the image (sections are called tiles or window). In each window, the image histogram intensities with highest frequencies are extended to the full data spectrum (in 8-bit images from 0 to 255) (figure 14). High exposure (high intensity values) and low exposure sections are optimized independently. CLAHE ameliorate the likelihood of identifying valid vectors by $4.7 \pm 3.2\%$ [30].

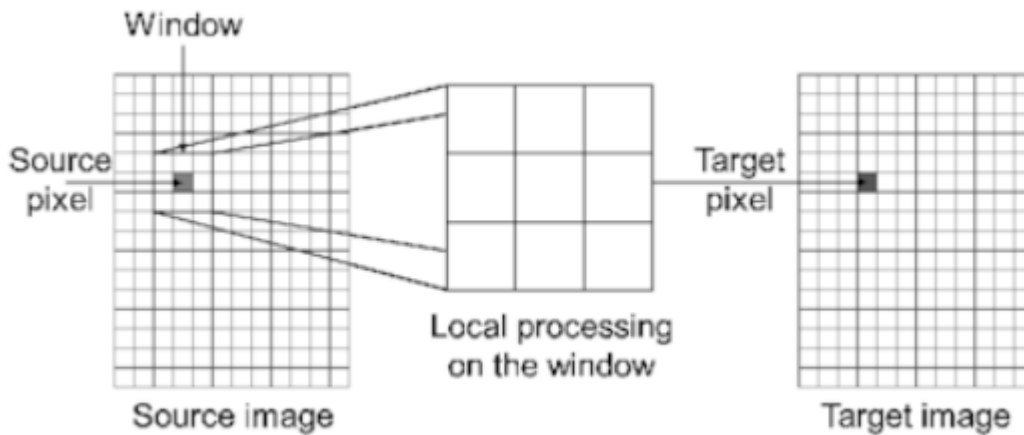


Figure 14: Noise reduction process [40]

(b) Intensity high-pass:

High-pass filter maintain the high-frequency information from the particle enlightenment, mostly used for inhomogeneous lighting which cause low-frequency background information. The filter accentuates the image element data and burkes all image low-frequency data (including all information on low-frequency movement) [28].

(c) Intensity capping:

The PIV technique presumes the same motion for all particles in the examination window. As perfectly uniform flow probably not exists, this will not be the case in reality. Bright points or dots in the window numerically share more to correlation signal that result in non-uniform flow [30]. This problem is avoided by intensity capping filter. The upper boundary of the intensity of grayscale is selected, and this upper limit replaced all pixels exceeding the threshold. Preventing the possible negative effect of image adjustments, only a slight quantity of the pixel intensity information is attuned in intensity capping unlike CLAHE [30]. The prospect of finding valid vector is enhanced by $5.2 \pm 2.5\%$ by intensity capping [30].

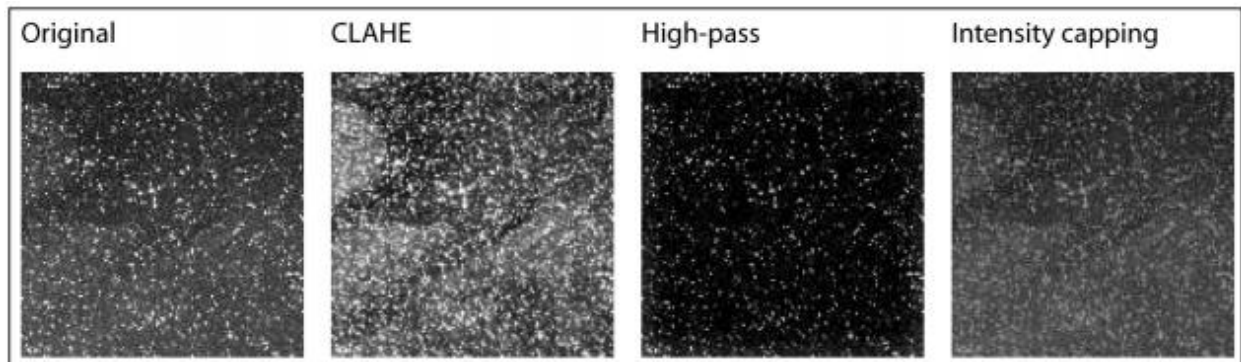


Figure 15: The consequence of pre-processing techniques [28]

2.5.3. Calibration

The image analysis in PIVlab calibration was required. Without calibrating the analysis, software will measure displacement vectors in pixels/second units. To calibrating an image, two measurements are required; Length of background and time between frames (that can be calculated using frames per second speed, also shown calculation in chapter 4).

For 30 fps video, time is calculated in the following way:

$$t = \frac{1}{30}$$

$$t = 33.3 * 10^{-3} \text{ sec} = 33.3 \text{ msec}$$

2.5.4. Computation of the cross-correlation function

Most commercially available data processing software is based on cross-correlation function Fast-Fourier Transform (FFT) algorithms. Most of these software are iterative and primarily grounded on an initial evaluation of velocity vectors. They use large interrogation windows related to a high SNR owing to the huge quantity of particles taken into account in the statistics, but compromised with poor spatial resolution. The dimensions of the interrogation window is also gradually reduced in practice to reach a final size of 32x32 or 16x16 pixels that depends on the density of particle images.

Typically raw images undergo pre-processing before the application of cross-correlation computation to improve contrast, decrease background noise (e.g. generated by particles deposits on windows), or to standardize the intensities of the particle to evade the bias of the cross-correlation function towards bulky particles.

2.6. Divergence

Divergence is the quantity of flux inflowing or exiting a point or the quantity diverging from a given volume. This could be defined as the rate of flux enlargement (positive divergence) or flux shrinkage (negative divergence). For example, If \mathbf{A} is velocity of the fluid, if $\nabla \cdot \mathbf{A}$ (divergence of vector field \mathbf{A}) is positive at a point, it means the expansion of fluid at a point or density at that point is falling with time. If $\nabla \cdot \mathbf{A}$ is negative, it implies that either the fluid is shrinking and its density is increasing at the point or the point is the sink of the fluid [41].

$$Divergence = \frac{Flux}{Volume} \quad \text{Equation no. 3}$$

Divergence of a vector quantity is a scalar [42].

- Positive divergence value implies the expansion of vector.
- Negative divergence value suggests the vector is converging.

Chapter 3: Experimental Setup

3.1. Experiment and required resources

In this chapter, an experimental set up was made with necessary instrumentation to evaluate the flame propagation using BOS (background oriented schlieren) and PIV (particle imaging velocimetry). **Figure 16** presents a comprehensive overview of experimental setup; details of each step have been described below.

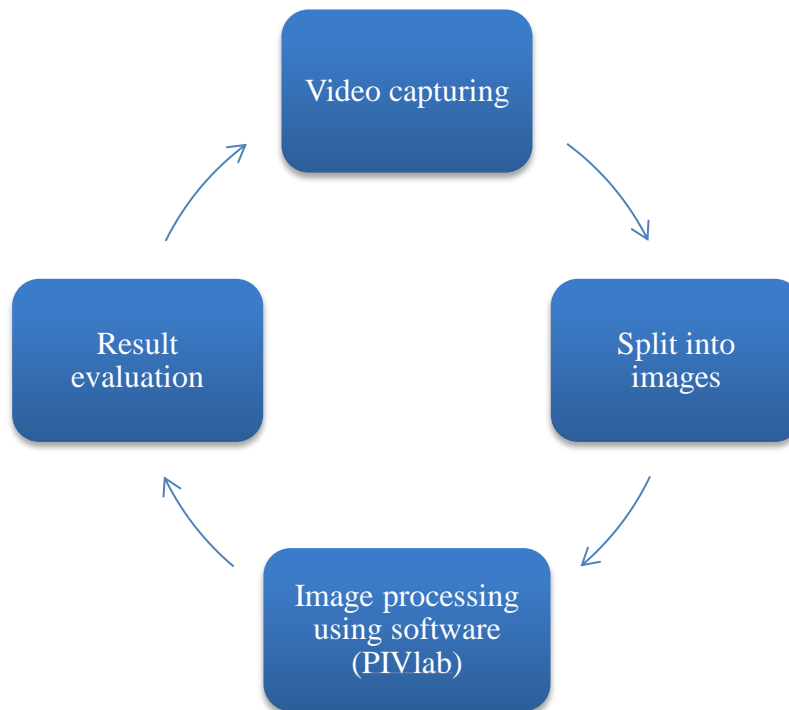


Figure 16: Experiment flow diagram

3.1.1. Video capturing

A smartphone Samsung Galaxy S8 SM-G950F was used to capture the videos for experiment. This smartphone has latest Android 9 operating system (OS). This smartphone is not the latest available model, but one of the recent smartphones with high resolution camera technology. This smartphone has main camera of 12-megapixel Dual pixel PDAF (Phase detection autofocus) with aperture of $f/1.7$. It can record videos at 3 different resolutions with different frames per second (2160p@30fps, 1080p@60fps, 720p@240fps).

a) Accessories:

- A premixed gas (propane-butane) burner was used.
- Two spot light were used to illuminate the background.
- Tripod for stability.

b) BOS background program:

Background is a random dot pattern image generated by a computer and printed on an A3 or A4. The pattern image was produced using the “makebospattern.m” file, which is integrated in PIVMat Toolbox (i.e. also included in **appendix A**) [34]. In **Table 2**, a comparison of number of dots generated and diameter for background pattern has been presented.

Table 2: Comparison between number of dots and diameter for background pattern

No. of Dots ‘n’	Diameter ‘d (mm)’	Remarks
50000	0.25	Very low visibility
75000	0.25	Very low visibility
100000	0.25	low visibility
125000	0.25	low visibility
150000	0.25	Good visibility
50000	0.5	low visibility
75000	0.5	low visibility
100000	0.5	Good visibility
125000	0.5	Good visibility
150000	0.5	Very Good visibility (Best choice for analysis)

From the above comparison, it could be concluded that best option for analysis is to used $n = 150000$ and $d = 0.5$ mm, to visualize the displacement of background.

3.1.2. Video split program/ software

Each animation or video consists of a series of still images. Then, several times a second, these images are played one by one, which fools the eyes into thinking that the object moves. The faster the images are played the more fluid and smoother the movement appears.

Each image is referred to as a frame that shows the term fps (frames per second). A video file on a computer simply stores all the frames together and plays them in order, and hundreds of thousands of the total frames stored for a typical movie. It's pretty easy to capture an image of one or two frames, and you have to pause the video and press the Print Screen key. It is incredibly inefficient and time consuming to capture images one by one, if you want to extract a range or succession of frames or even all frames from a short video. For that purpose, a program is needed that can extract frames from a video and automatically save them as image files, such as jpg, png or bmp. There are a lot of software's available for video split into images (like VLC media player, VirtualDub etc.) and also this can be done on MATLAB script program. I have used MATLAB script program for video splitting (**appendix D**).

3.1.3. Data processing/BOS image processing software

After receiving the image pairs to a computer, BOS processing was done and then processed through PIV image-correlation software. There are a lot of different software's developed for image processing. One of them is ImageJ which is developed by the US NIH (National Institutes of Health) [36], but a smarter GUI (Graphical User Interface) based BOS and PIV image analysis program is PIVlab, which has been used for current work [28].

Beside PIV analysis, a MATLAB script program was written to get absolute difference between distorted and reference without disruption images (**appendix C**). From this analysis, we can get flame structure (figure 17).


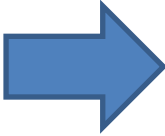

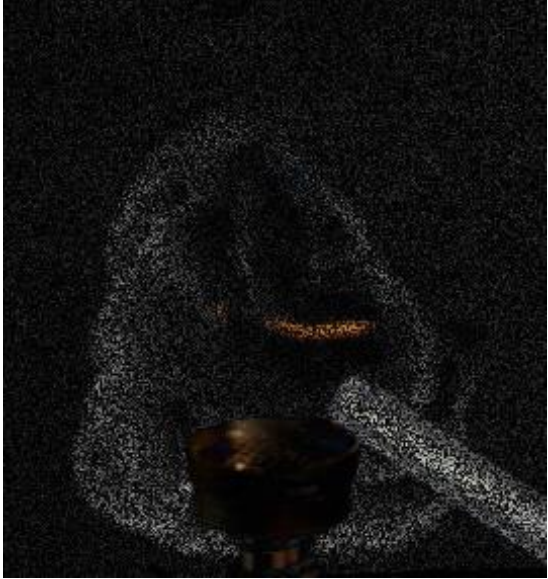
(a) Reference image (without disturbance)		
		
		
(b) Distorted image		(c) Resulted image by absolute difference

Figure 17: Figure describes inputs (left side) and output of MATLAB program

3.1.4. Results evaluation

Results have been discussed and evaluated in chapter 4.

3.2. Challenges:

Below is the list of few challenges faced during the experimental setup and experiment.

- Proper light distribution on background.
- Program to make random number of dot for background.
- Optimum background with sufficient number of dots and diameter to see displacement of background.
- Slow speed video camera (more frames per seconds) to gather more information from images.
- Calibration of images to get real unit data from the PIVlab analysis.
- Colors scale/legends on analyses images; color scale shows measure/degree of shift pixel between two exposures.

Chapter 4: Results & Discussion

This chapter describes and presents the analysis of experiments followed by a discussion on the research results. The results relate to the research questions that led the study. Data were used investigated to describe flame propagation using background oriented schlieren technique.

4.1. Artificial Experiment:

An artificial experiment was performed to check ability of software (PIVlab) to compute displacement vectors from input images. In this experiment, MATLAB program was used to produce images of dot background (black color) and seeding particles (red color). MATLAB script for seeding particles with dot pattern has been presented in **Appendix B**. Background dots were kept fixed and seeding particles were generated in the same way using rand function.

4.1.1. 0.01 pixel shift:

Figure 18 shows the raw images for 0.01 pixel shift produced by using MATLAB program.

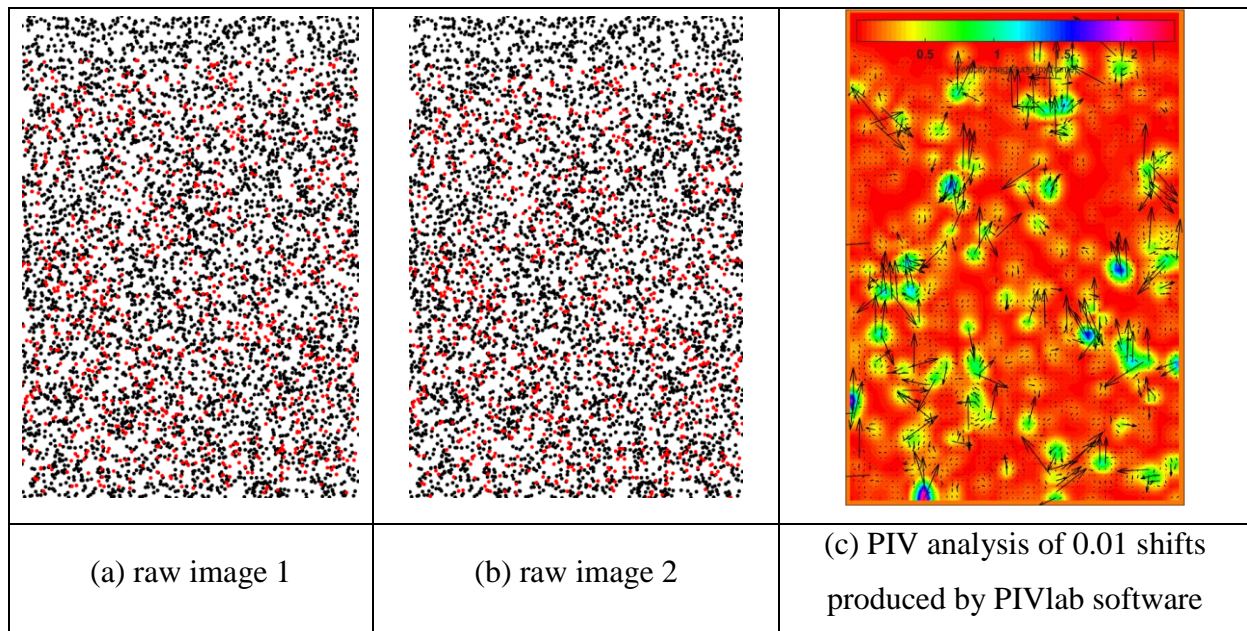


Figure 18: Artificial experiment for 0.01 pixel shift produced by using MATLAB program

As shown in **Figure 18**, (a) seeding particles in image have some randomly generated certain position $(x_i, y_i; i = 1 \dots N)$, and then all of particles shifted to a certain amount (0.01 pixel shift) of pixels along y-axis (in the same direction, upward) in **Figure 18**, (b). For that experiment, two images with same background and position of the seeding particles were experiencing 0.01 (figure 18) and 0.05 (figure 19) pixels shifts along y-axis.

4.1.2. 0.05 pixel shift:

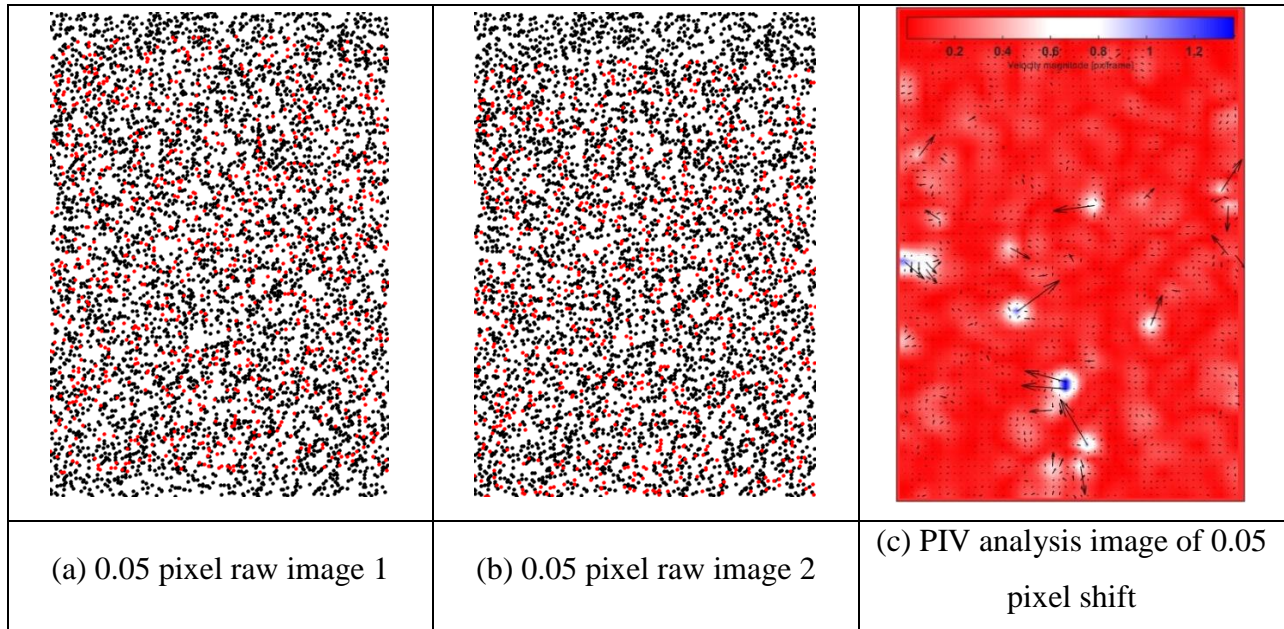


Figure 19: Artificial experiment for 0.05 pixel shift

In raw images BOS particles (black background dots) are still at the same place while seeding particles (red dots) shifted along y-axis, 0.01 and 0.05 pixels respectively. In **figure 18**, **(c) and 19**, **(c)**, all the seeding particles move along y-axis but in PIV analysis, it is apparent that some vectors are displaced in different directions. This disarray is due to the collision of BOS and seeding dots and it could be concluded that PIV system correlation has problems in determining what exactly the movement is.

In a way, the result was expected because BOS pattern is static and has zero displacement. When all the seeding particles which are moving upwards because of interaction with background dots/pattern (BOS particles) look like they move sideways or downwards. They

neglect real movement and showing high magnitude movement. From figure 18 and 19, we can analyze that when the pixel shift is small (0.01 pixel shift), more vectors are scattered in different directions as compared to 0.05 pixel shift.

4.2. Experiment without background pattern:

First, experiment using white paper was done i.e. without background pattern to check applicability of BOS process and to see the possibility of application of BOS method on white background. **Figure 20** portrays the input images to PIVlab (minimum two images to get displacement vectors) obtained using experiment.

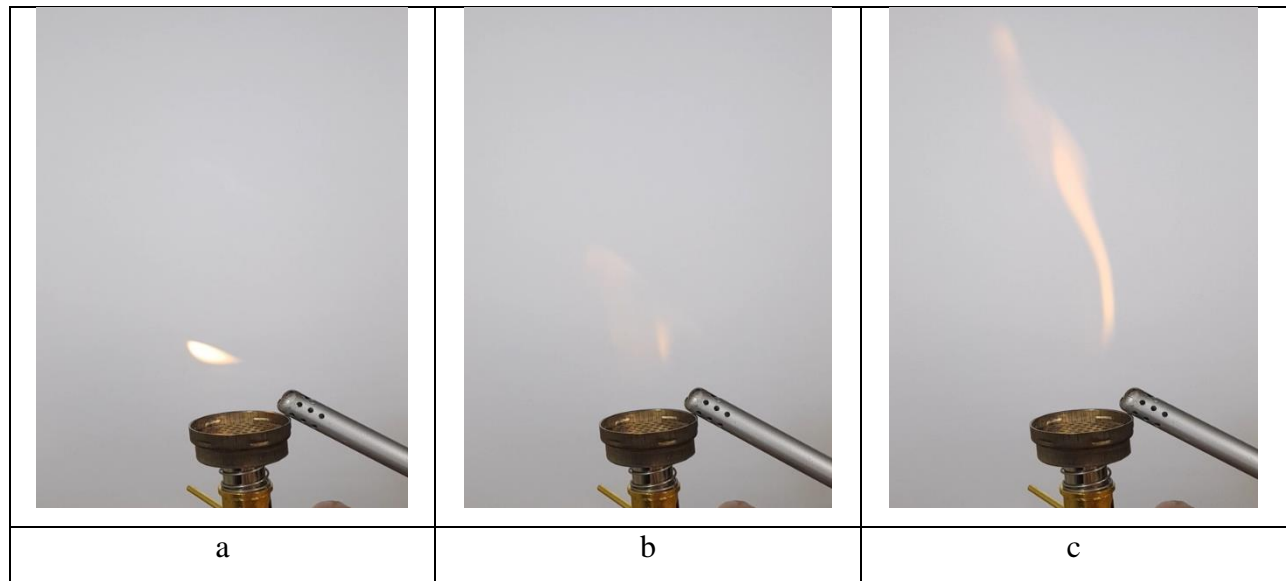


Figure 20: Input images of the experiment using white paper

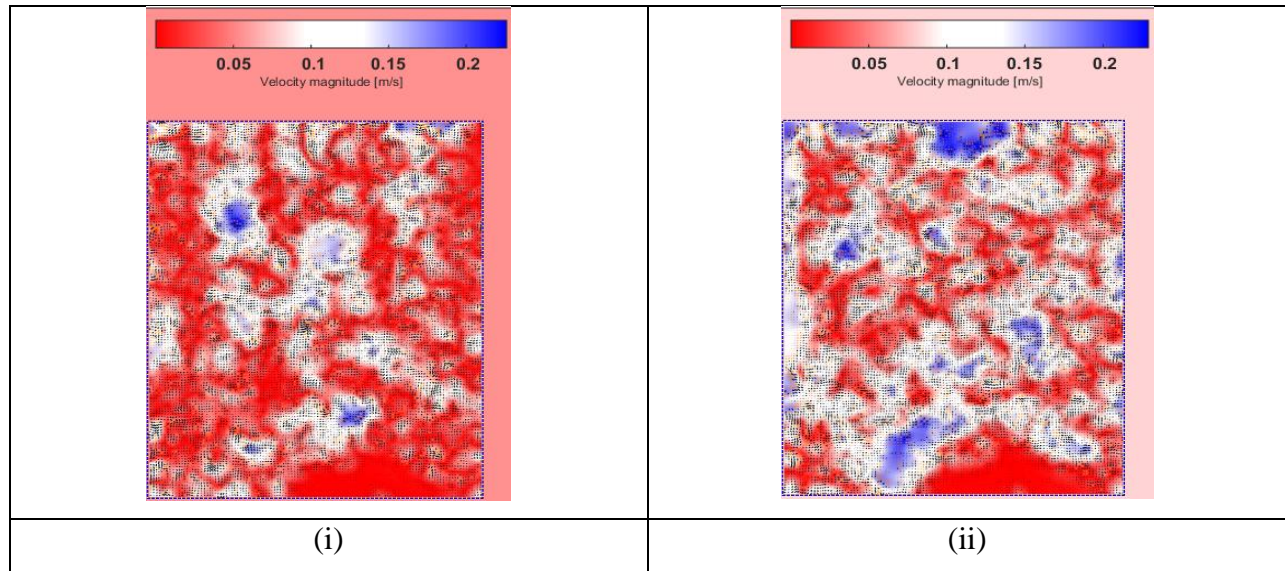


Figure 21: PIV analysis images of experiment on white paper

For one PIV analysis image, we need two input/raw images. Some important results that could be drawn from the **figure 21** are:

- Flame structure movements cannot be correlated because there's no background pattern behind the flame and no change in reflective index as BOS is based on reflective index.
- One of the advantages of background pattern is that we can correlate the structure using relative movement. If strong light will be used on object and looked on background (white paper), then there's possibility to get the shadow of the object. This process is very close to shadowgraphy which is out of scope of this study.

4.3. Experiment on BOS background:

An experiment was done for the study of flame propagation using background oriented schlieren technique (Figure 22). Images from the experiment then analyzed by particle imaging velocimetry using PIVlab software to obtain different parameters (like velocity magnitude, vorticity etc.) and structure of the flame. This experiment was performed by varying frames per second speed on camera (30 fps, 60 fps and 240 fps used).

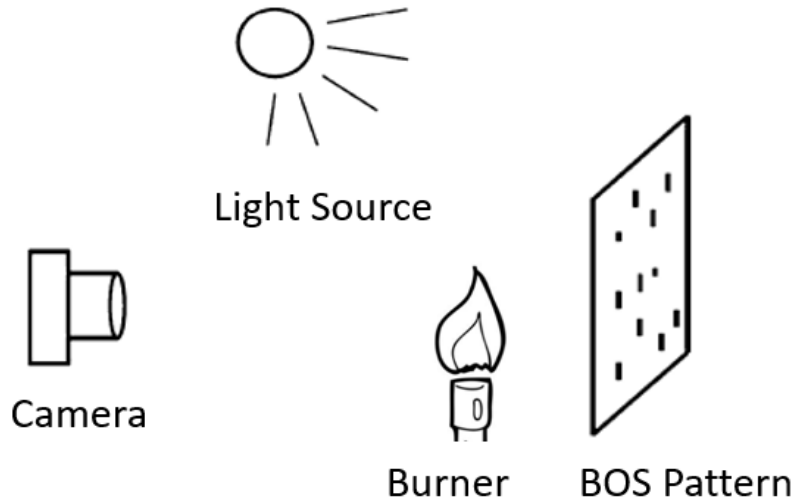


Figure 22: Schematic diagram of BOS experiment

4.3.1. 30 fps:

For 30 fps video, calibration was done using following time and length that can be measured from background:

$$t = \frac{1}{30}$$

$$t = 33.3 * 10^{-3} \text{ sec} = 33.3 \text{ msec}$$

Start ignition t = 0 (taken as reference for time):

Following are figures shown in sequence for every experiment, first at t = 0 then after 5 and 10 sec of ignition to portray development of flame structure and upward flow from start to fully developed at 30 fps.

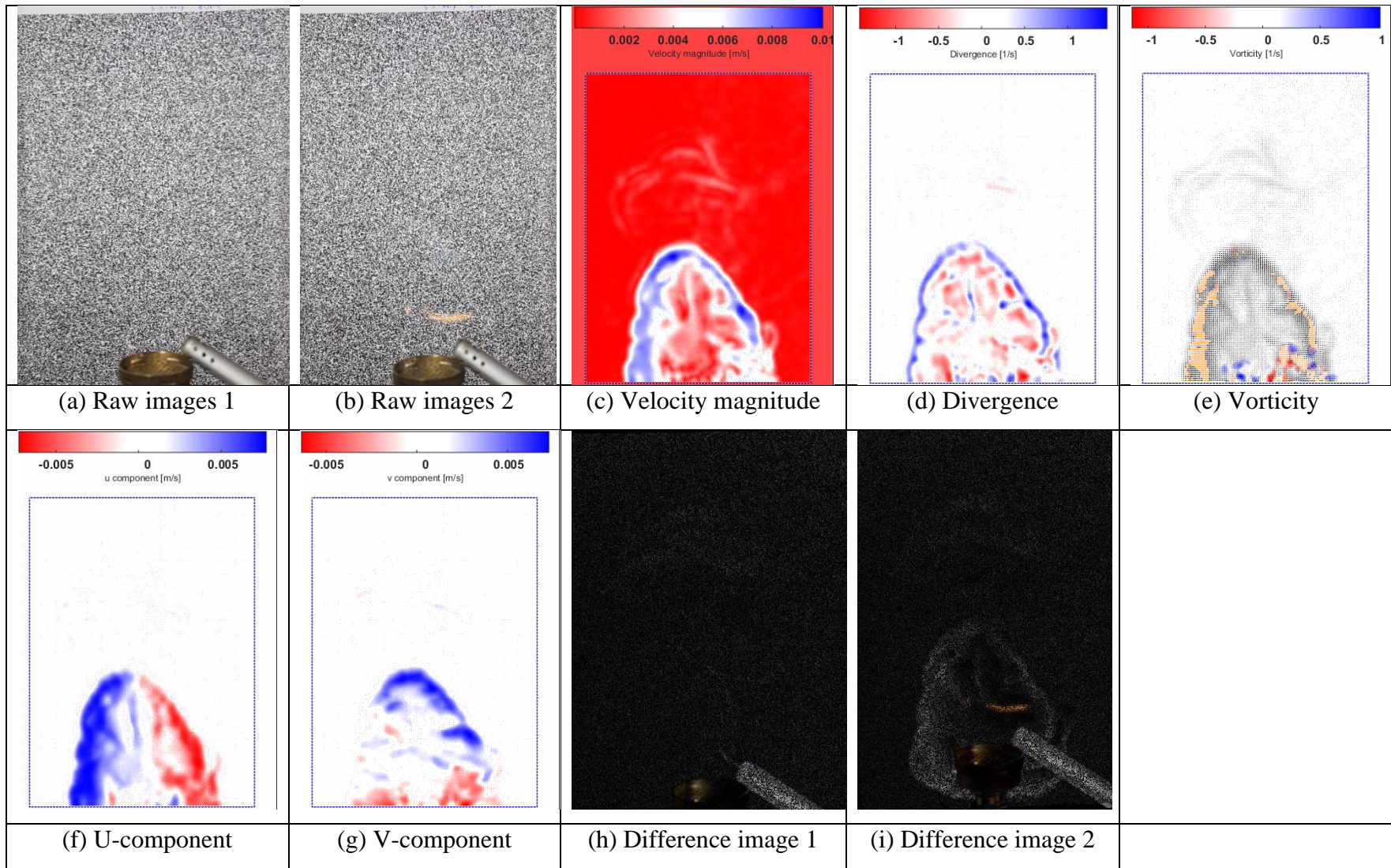


Figure 23: PIV analysis images 1 at reference time for 30 fps

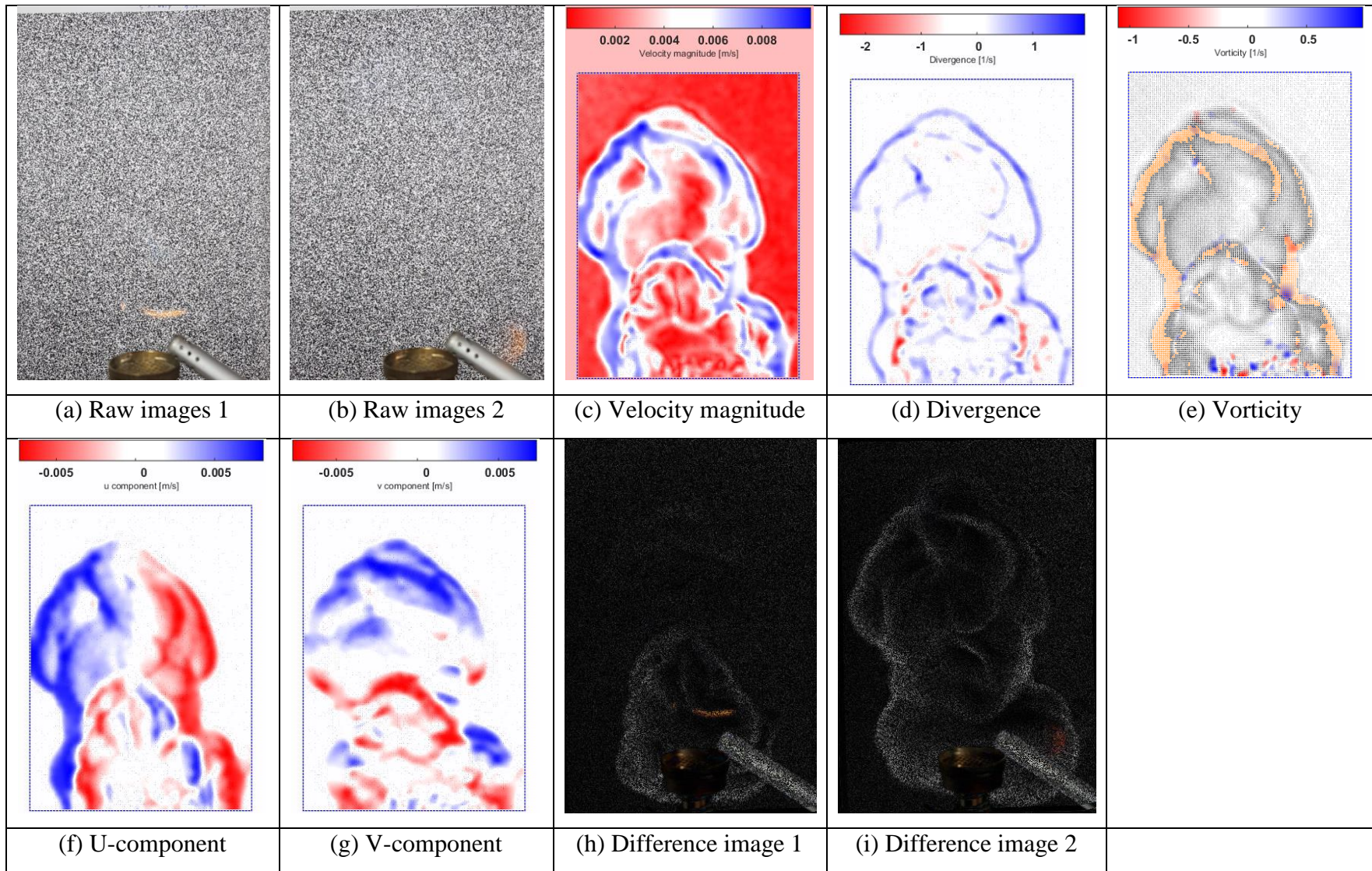


Figure 24: PIV analysis images 2 at reference time for 30 fps

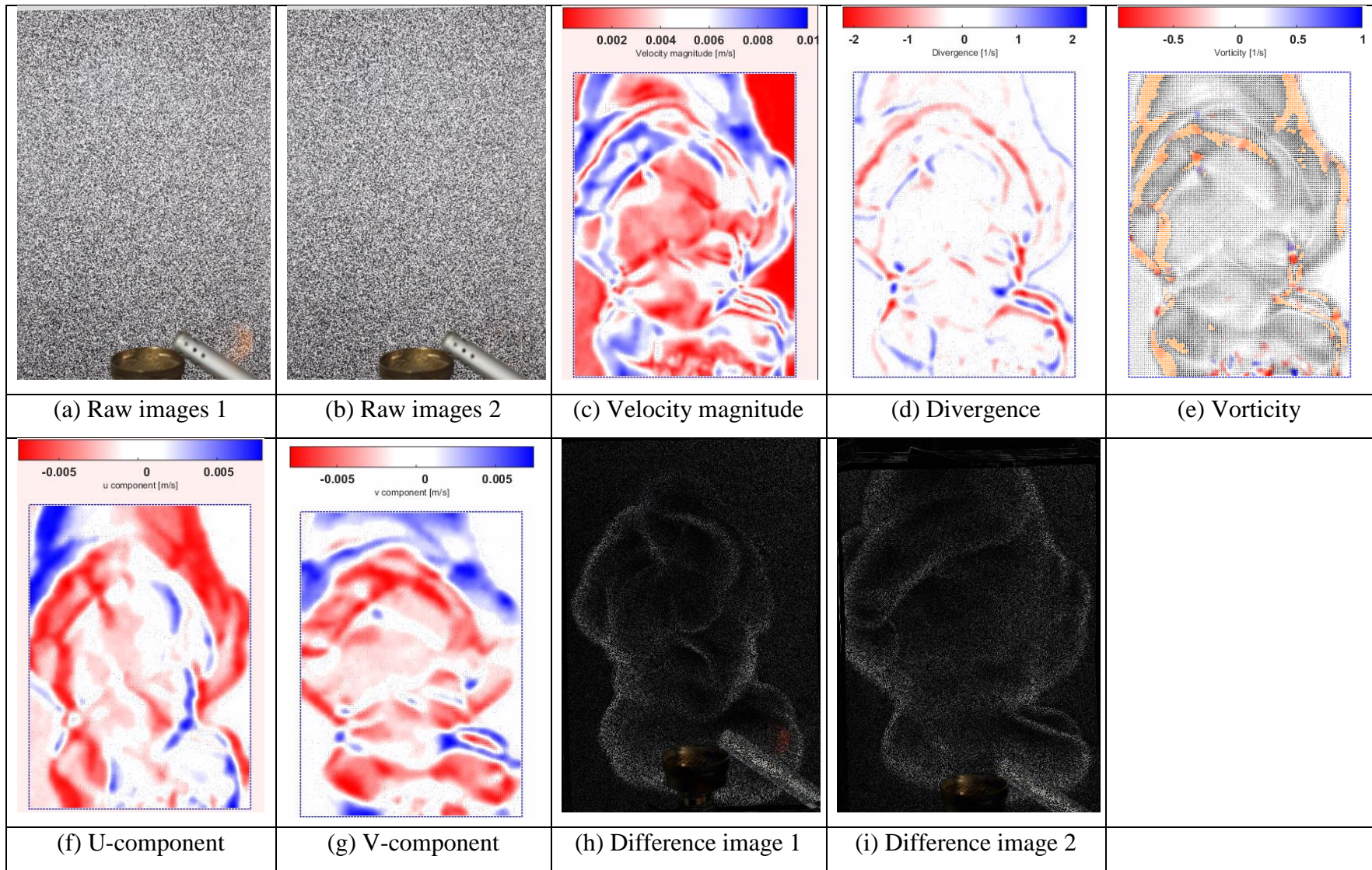


Figure 25: PIV analysis images 3 at reference time for 30 fps

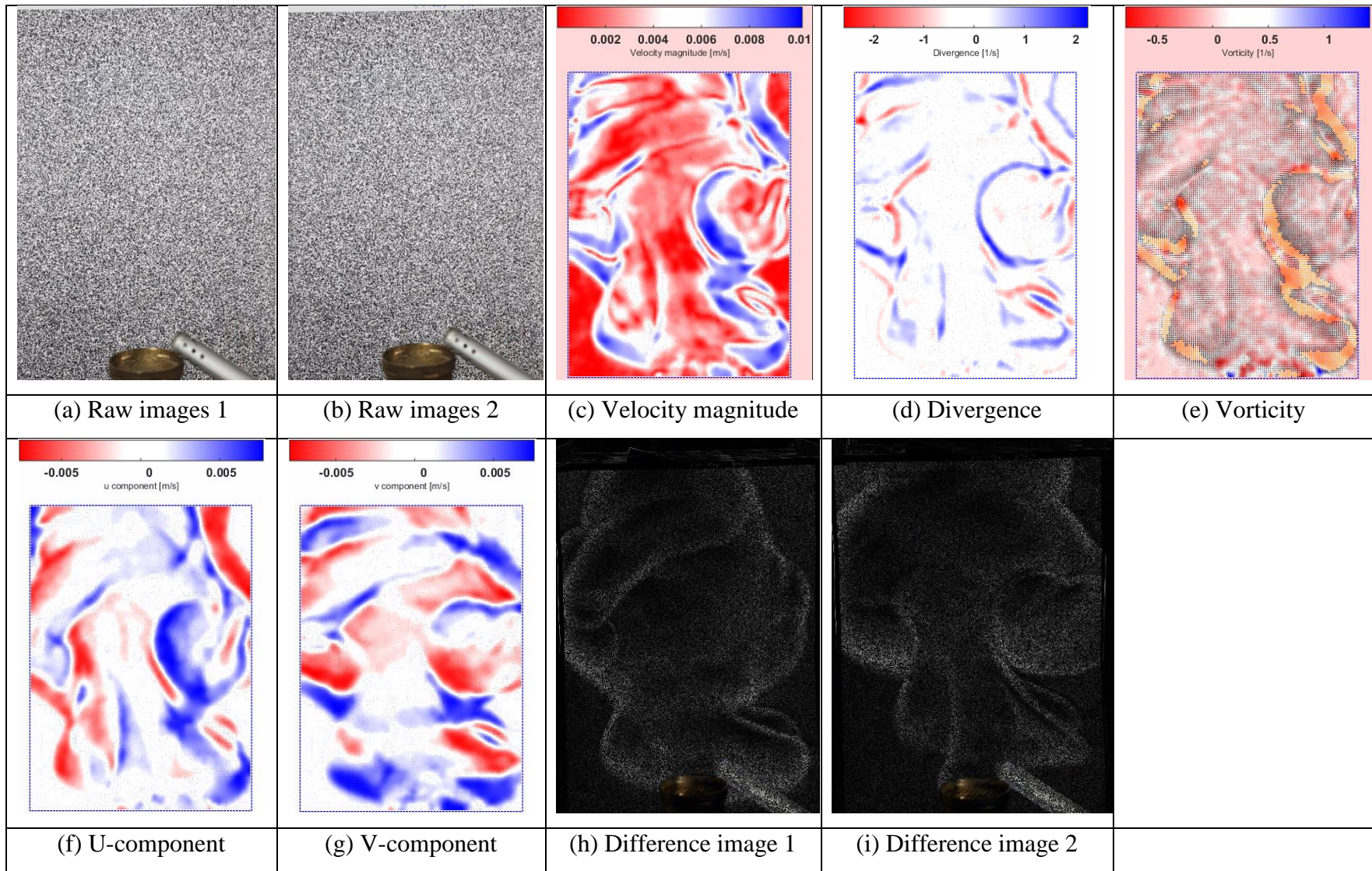
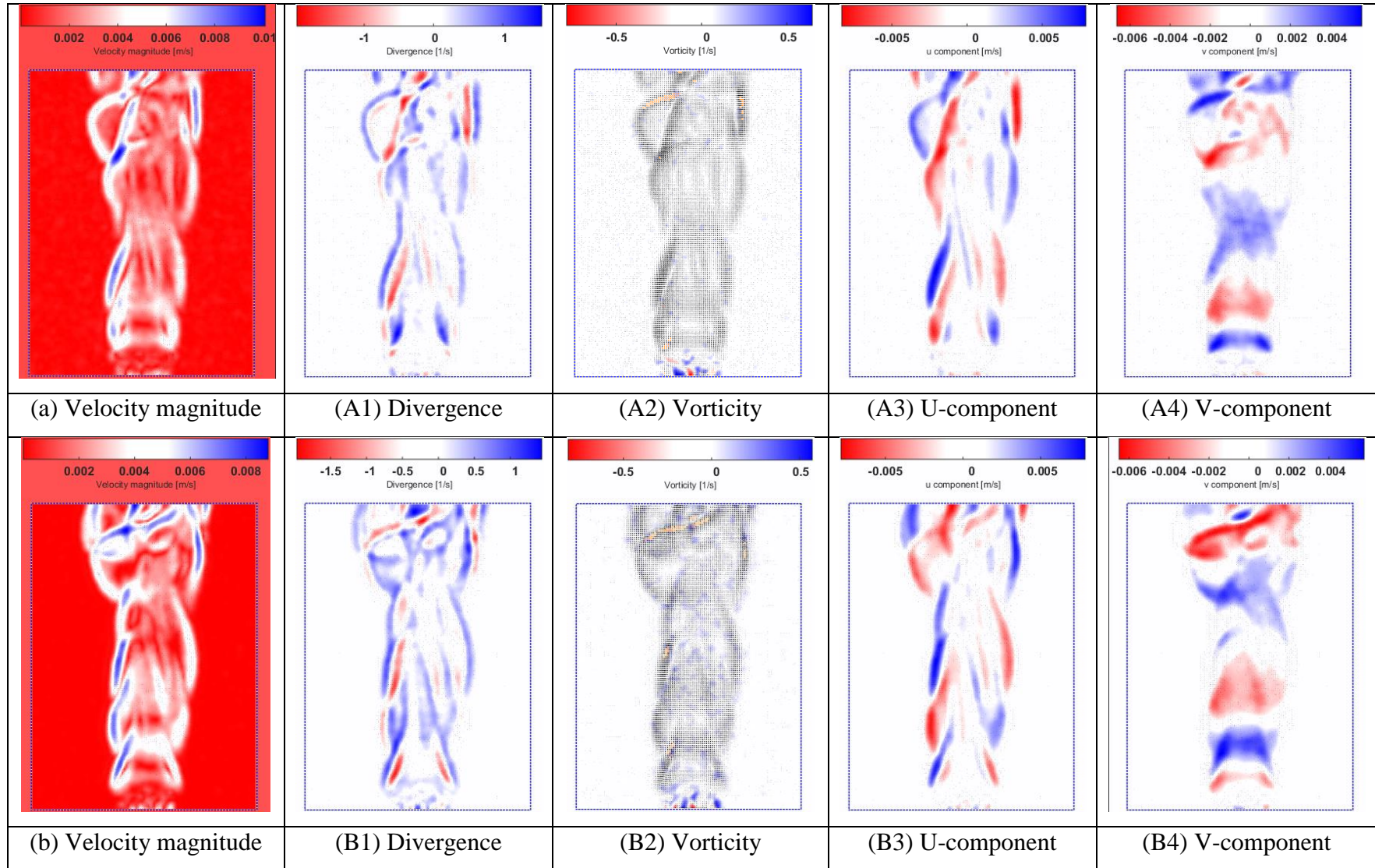


Figure 26: PIV analysis images 4 at reference time for 30 fps

After 5 sec from reference time:



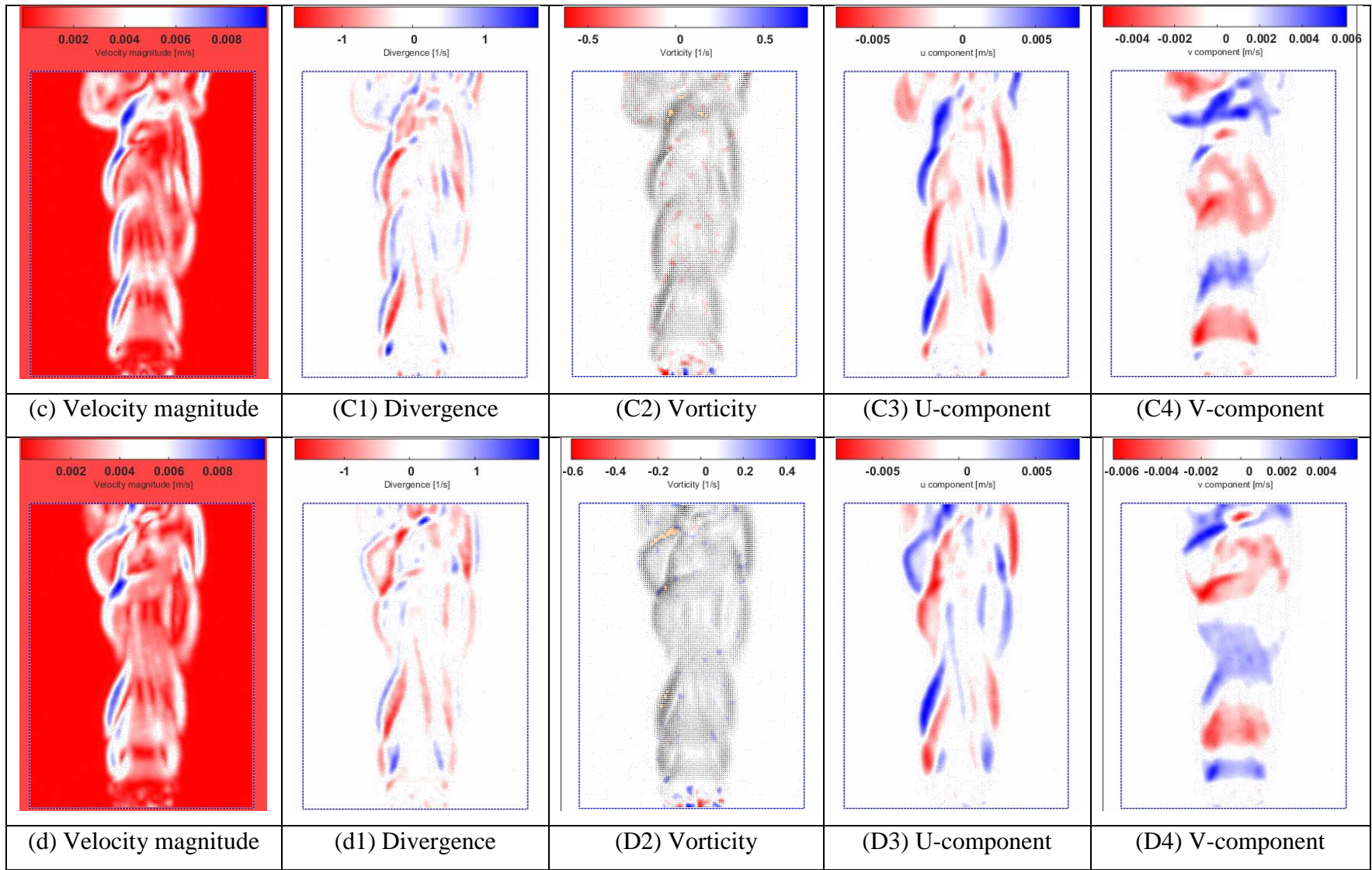
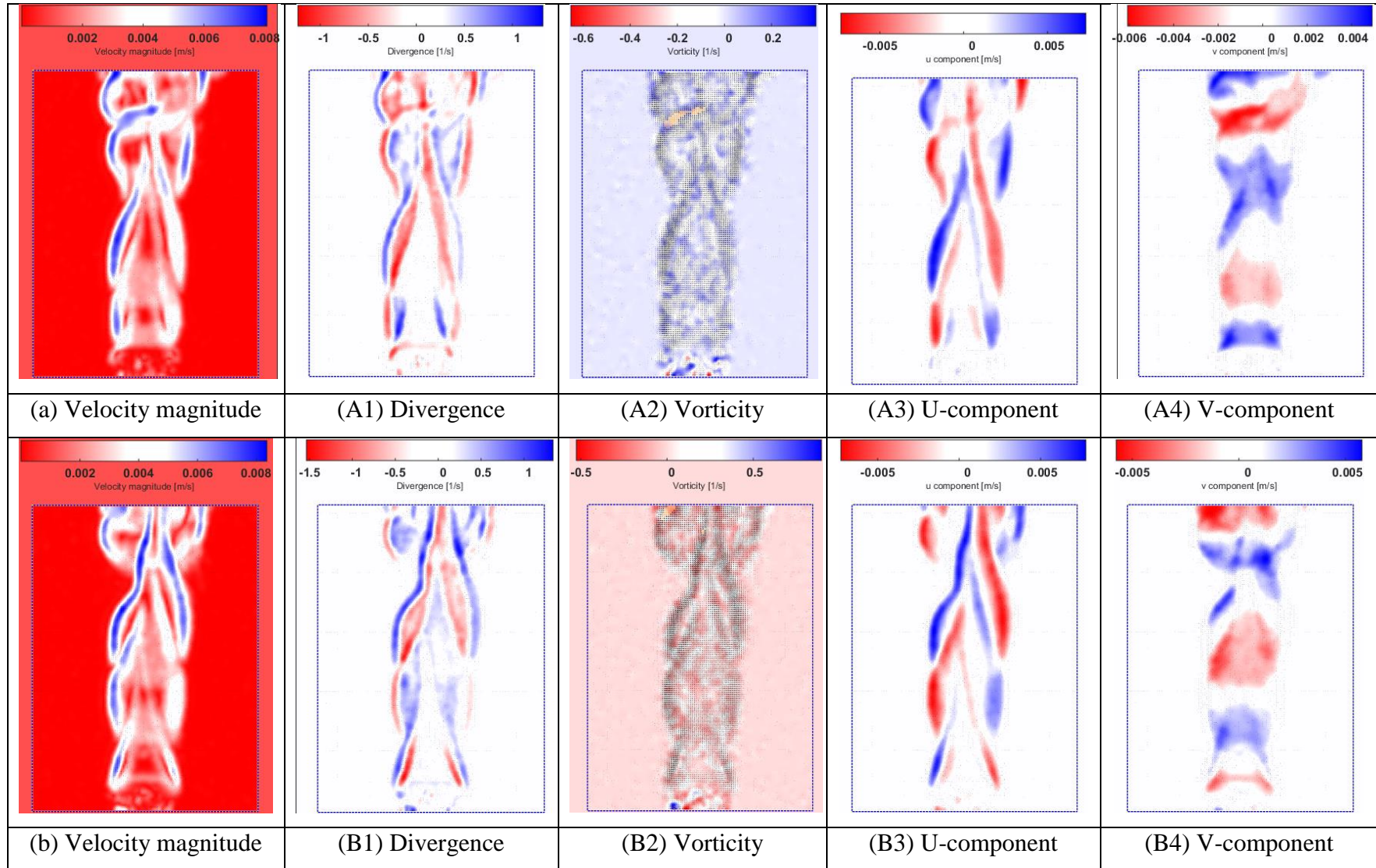


Figure 27: PIV analysis images after 5 sec for 30 fps

After 10 sec from reference time:



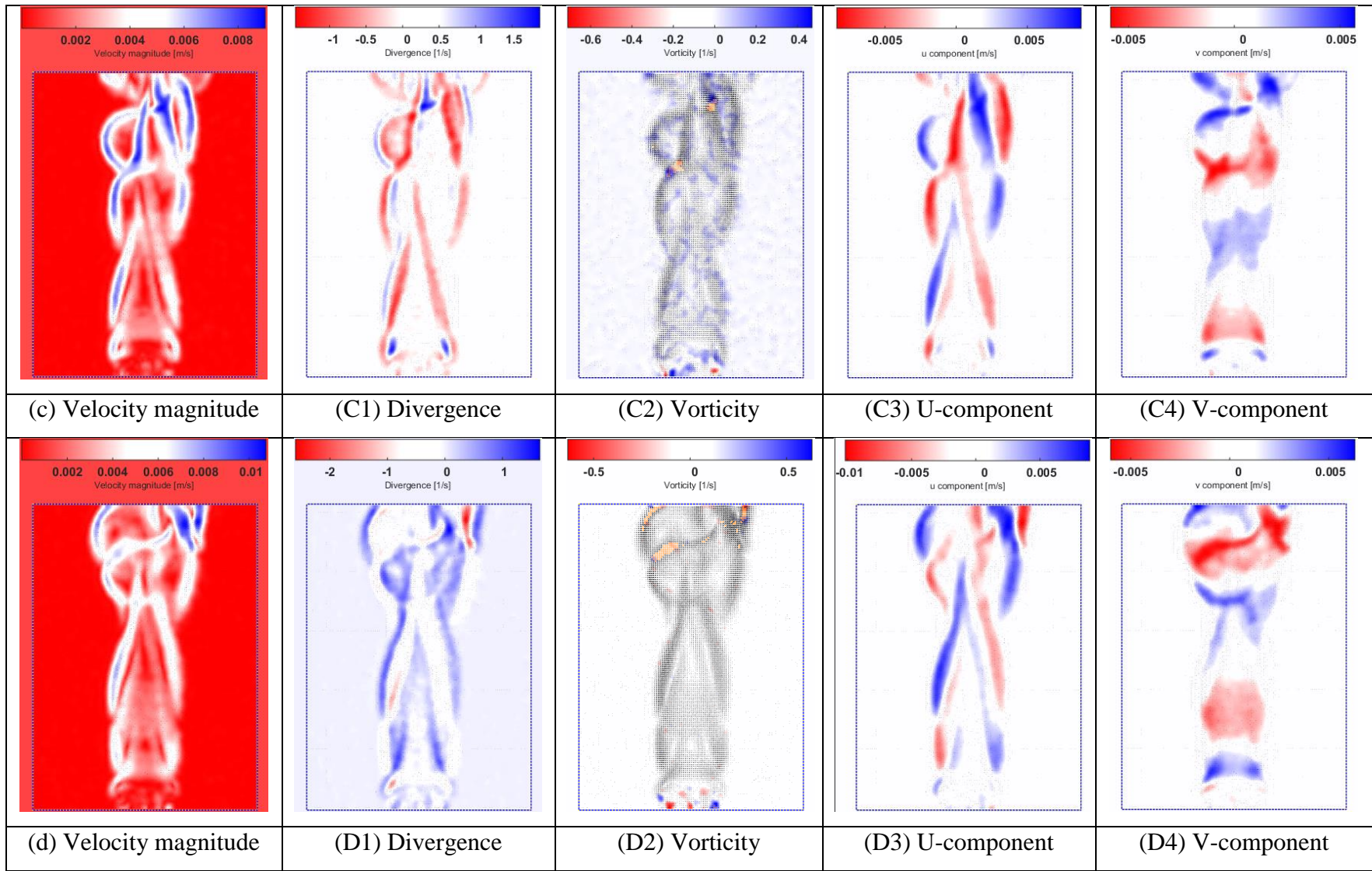


Figure 28: PIV analysis images after 10 sec for 30 fps

4.3.2. 60 fps (Start ignition t = 0):

For 60 fps video, calibration was done using following time and length that can be measured from background:

$$t = \frac{1}{60}$$

$$t = 16.667 * 10^{-3} \text{ sec}$$

$$t = 16.667 \text{ msec}$$

Following are figures shown in sequence, first at t = 0 then after 5 and 10 sec of ignition to reveal progress of flame flow from burner tip to top at 60 fps and flame structure.

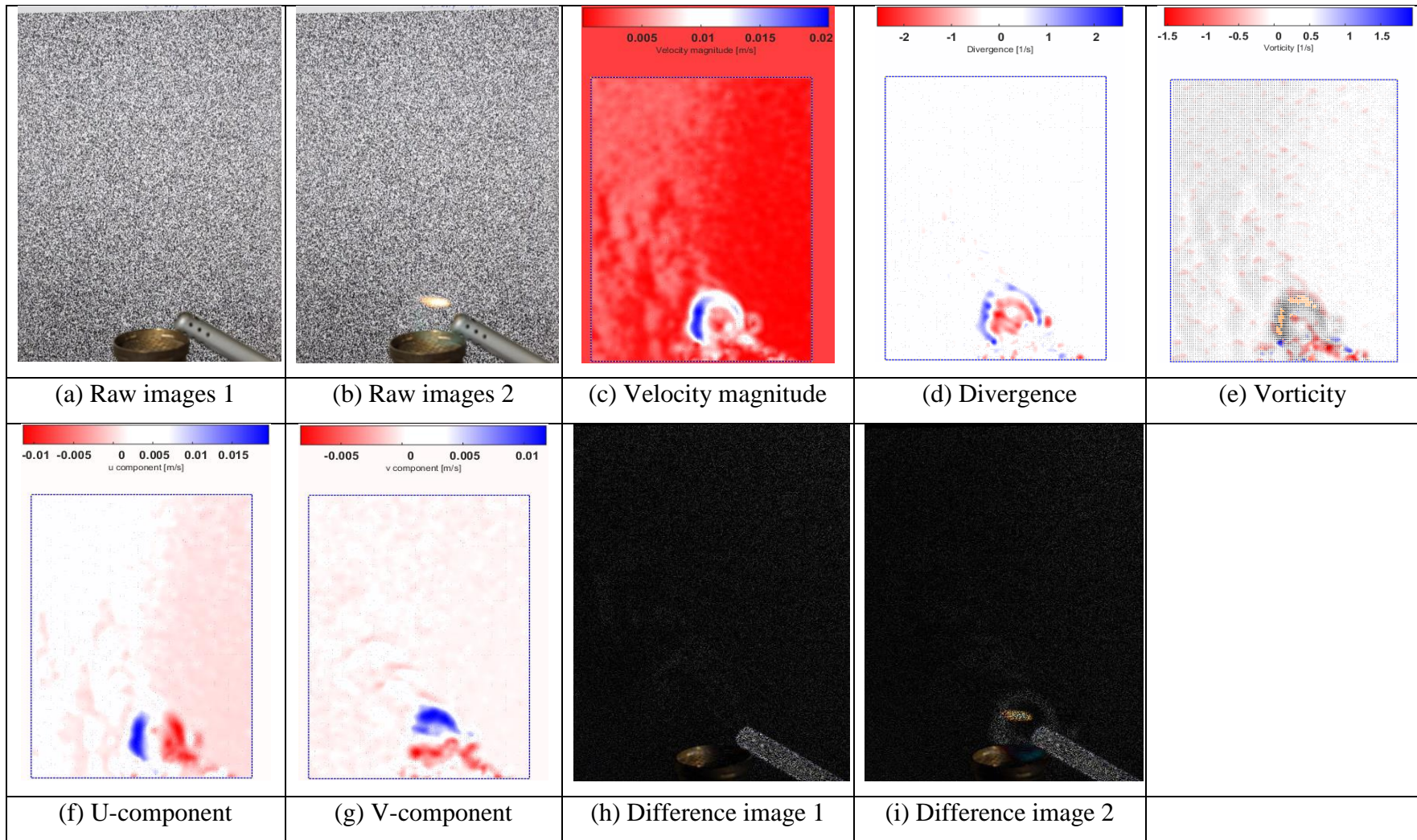


Figure 29: PIV analysis images 1 at reference time for 60 fps

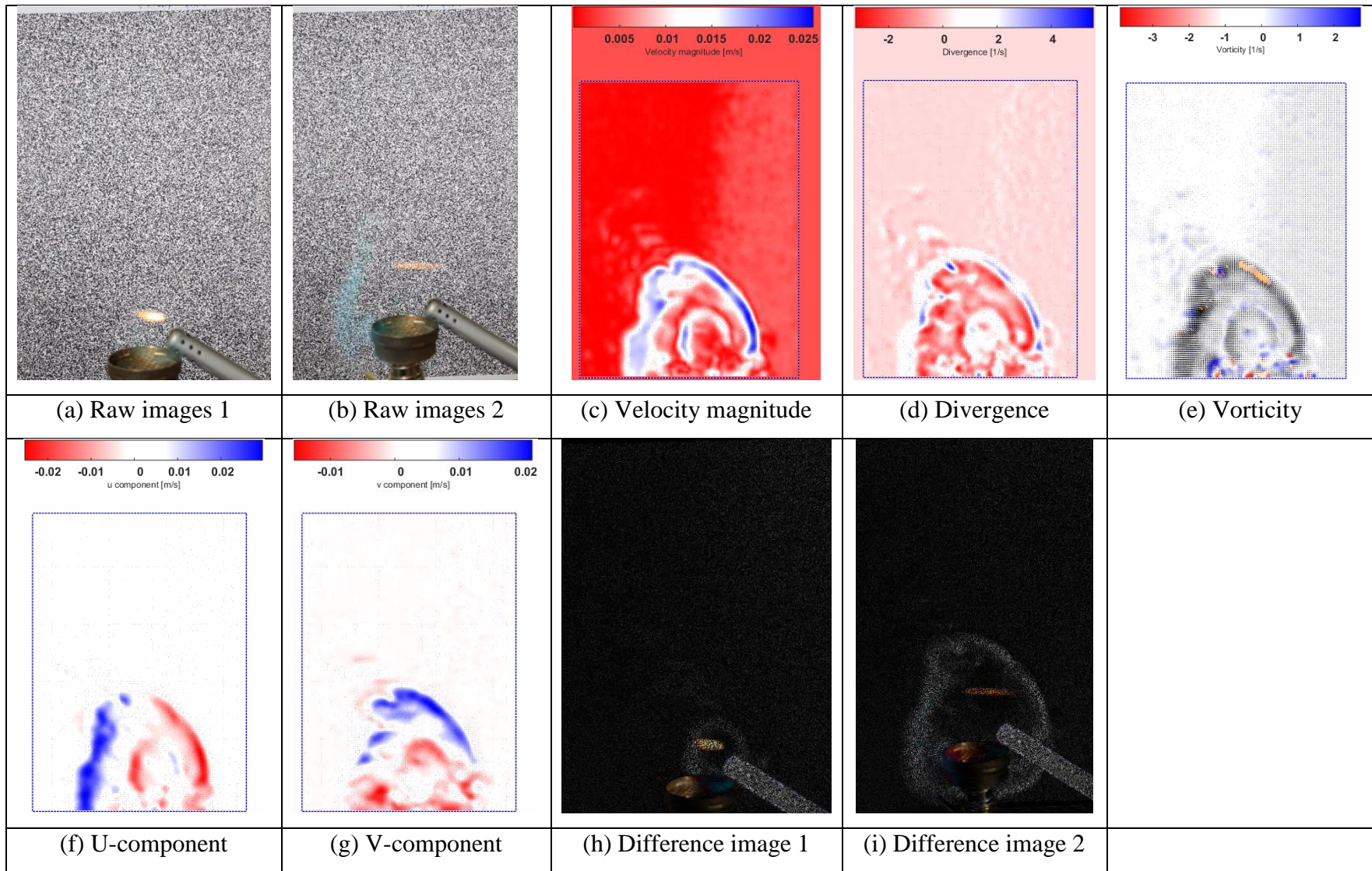


Figure 30: PIV analysis images 2 at reference time for 60 fps

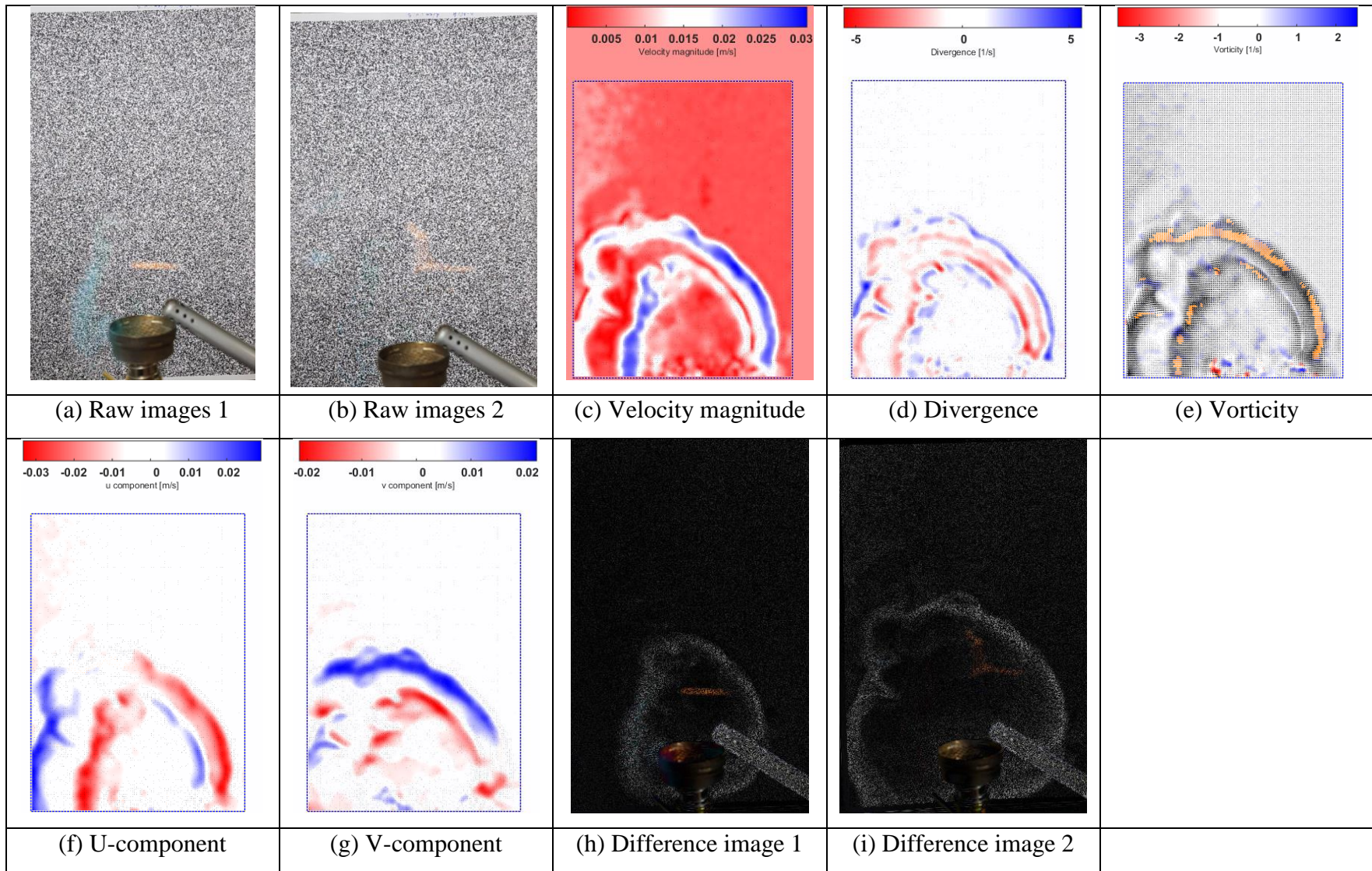


Figure 31: PIV analysis images 3 at reference time for 60 fps

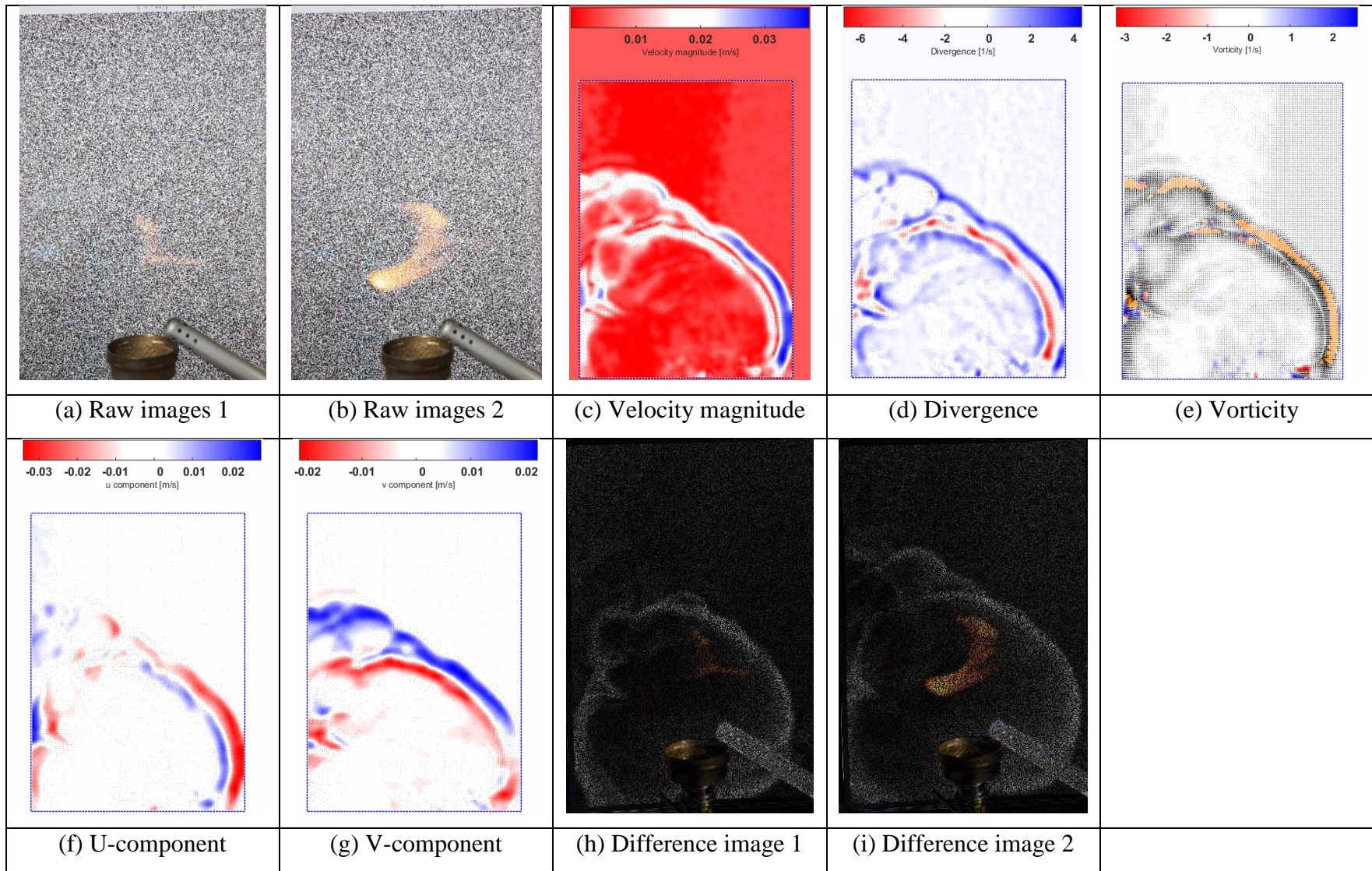
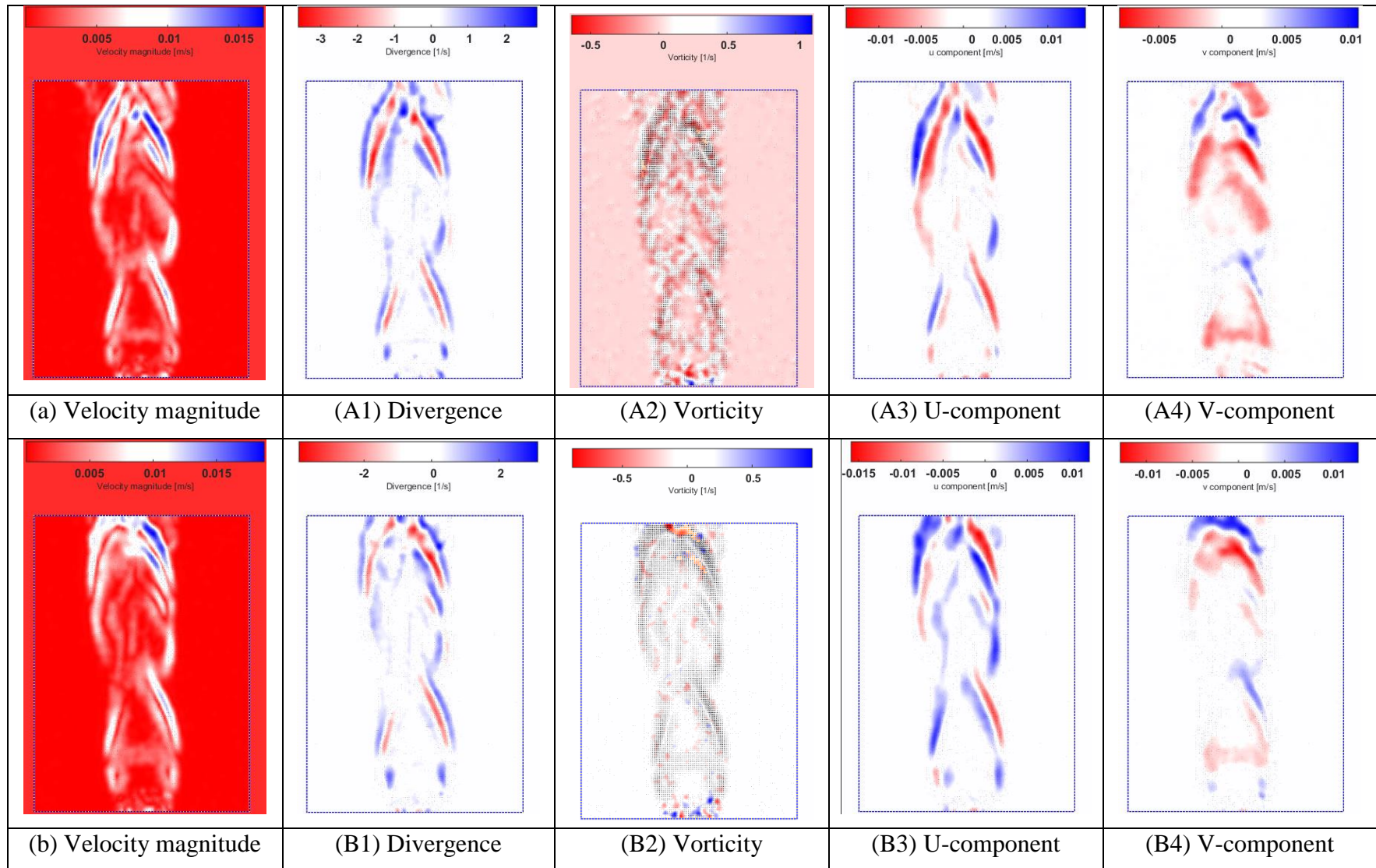


Figure 32: PIV analysis images 4 at reference time for 60 fps

After 5 sec from reference time:



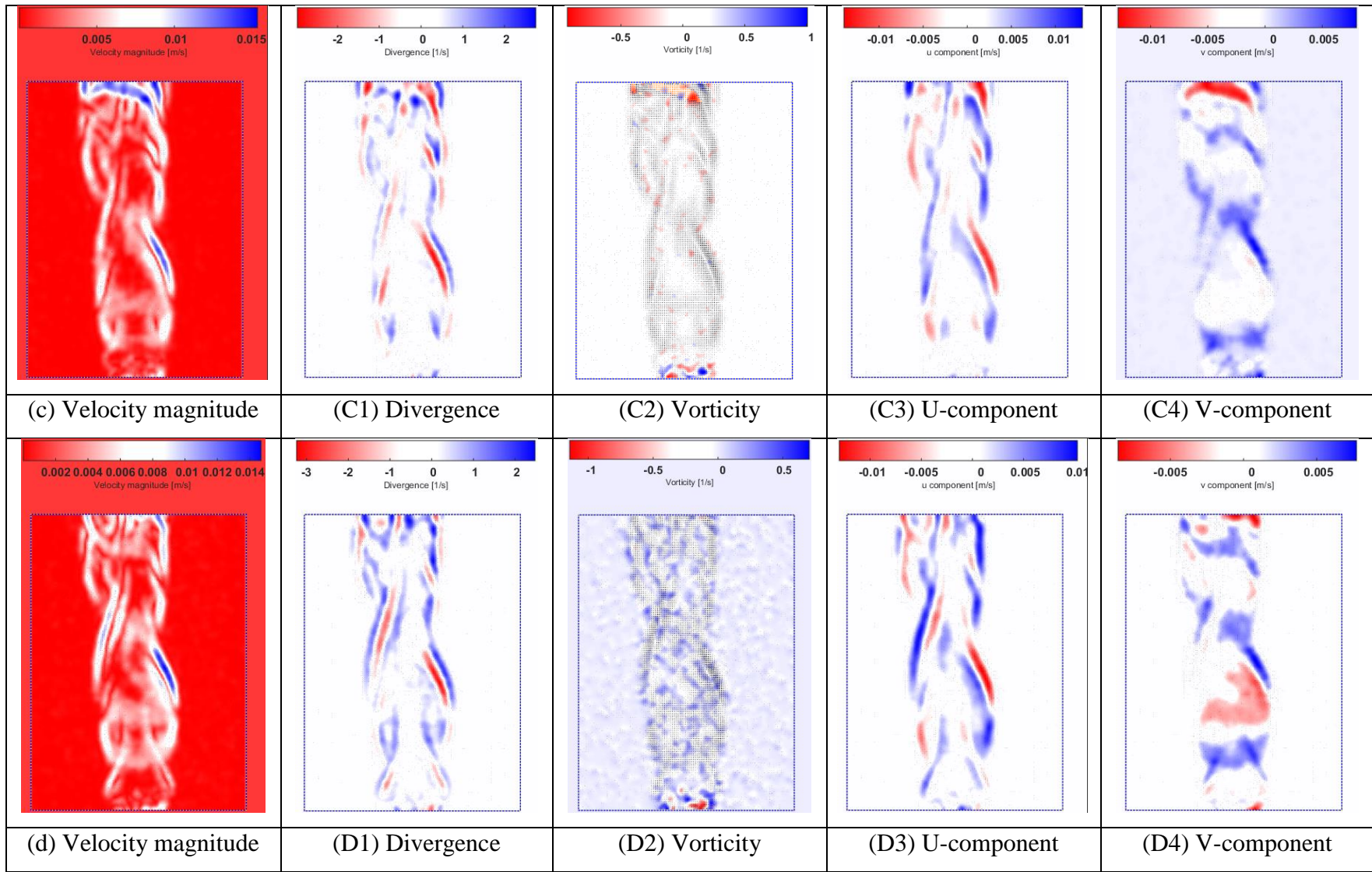
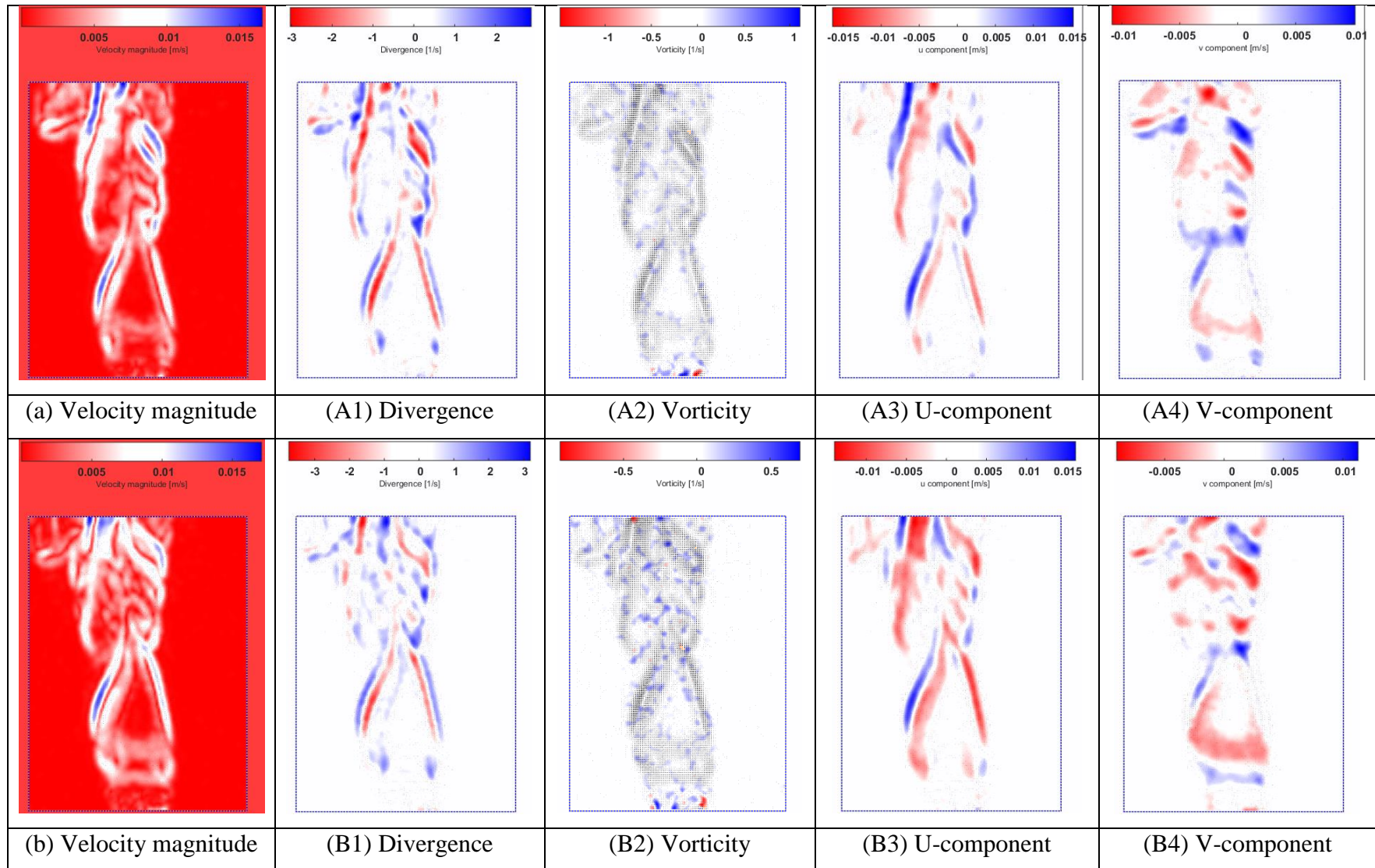


Figure 33: PIV analysis images after 5 sec for 60 fps

After 10 sec from reference time:



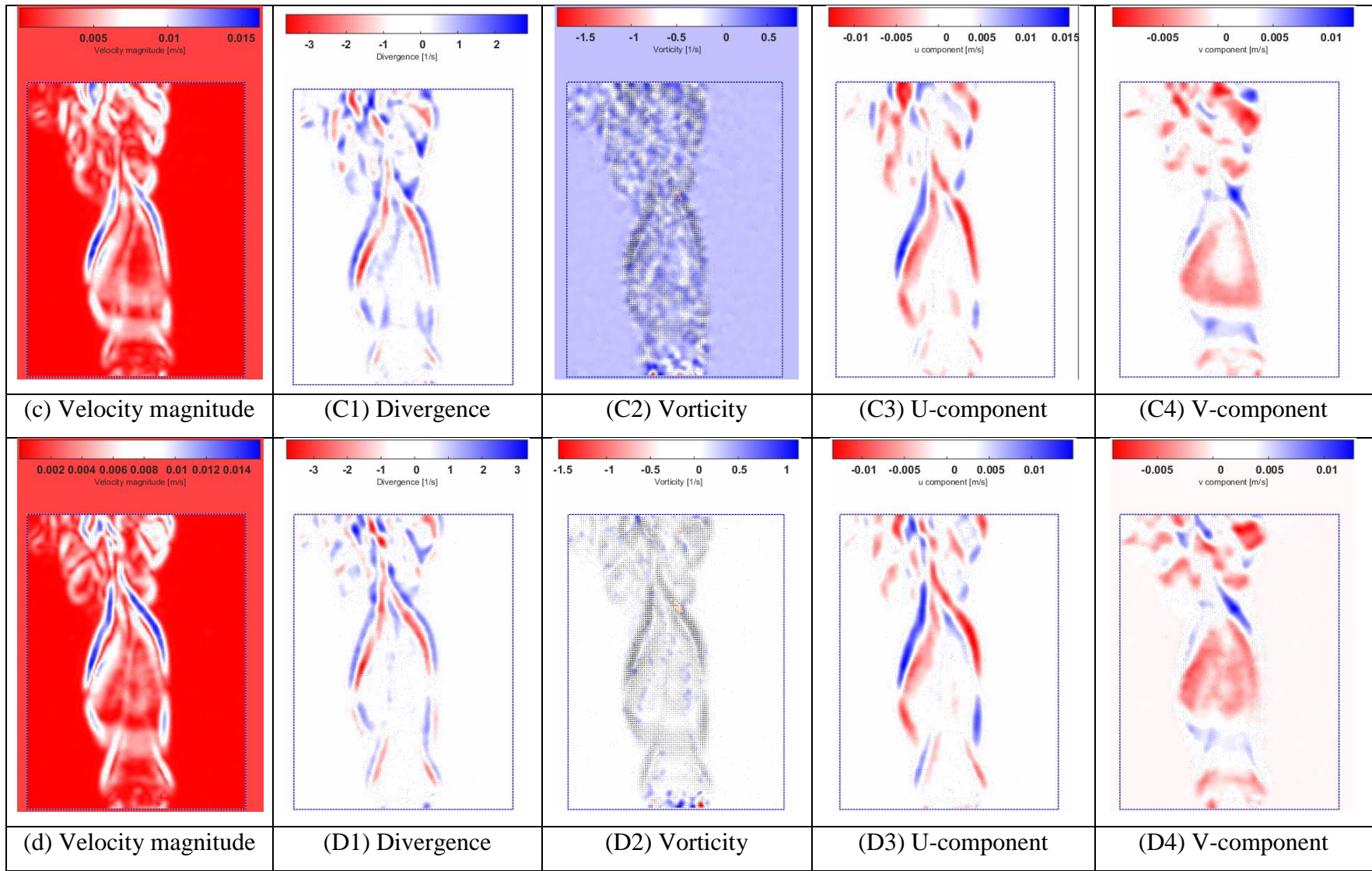


Figure 34: PIV analysis images after 10 sec for 60 fps

4.3.3. 240 fps (Start ignition t = 0):

For 240 fps video, calibration was done using following time and length that can be measured from background:

$$t = \frac{1}{240}$$

$$t = 4.163 * 10^{-3} \text{ sec}$$

$$t = 4.163 \text{ msec}$$

Following are figures shown in sequence, first at t = 0 then after 5 and 10 sec of ignition to display growth of flame flow from start to top at 240 fps and flame structure.

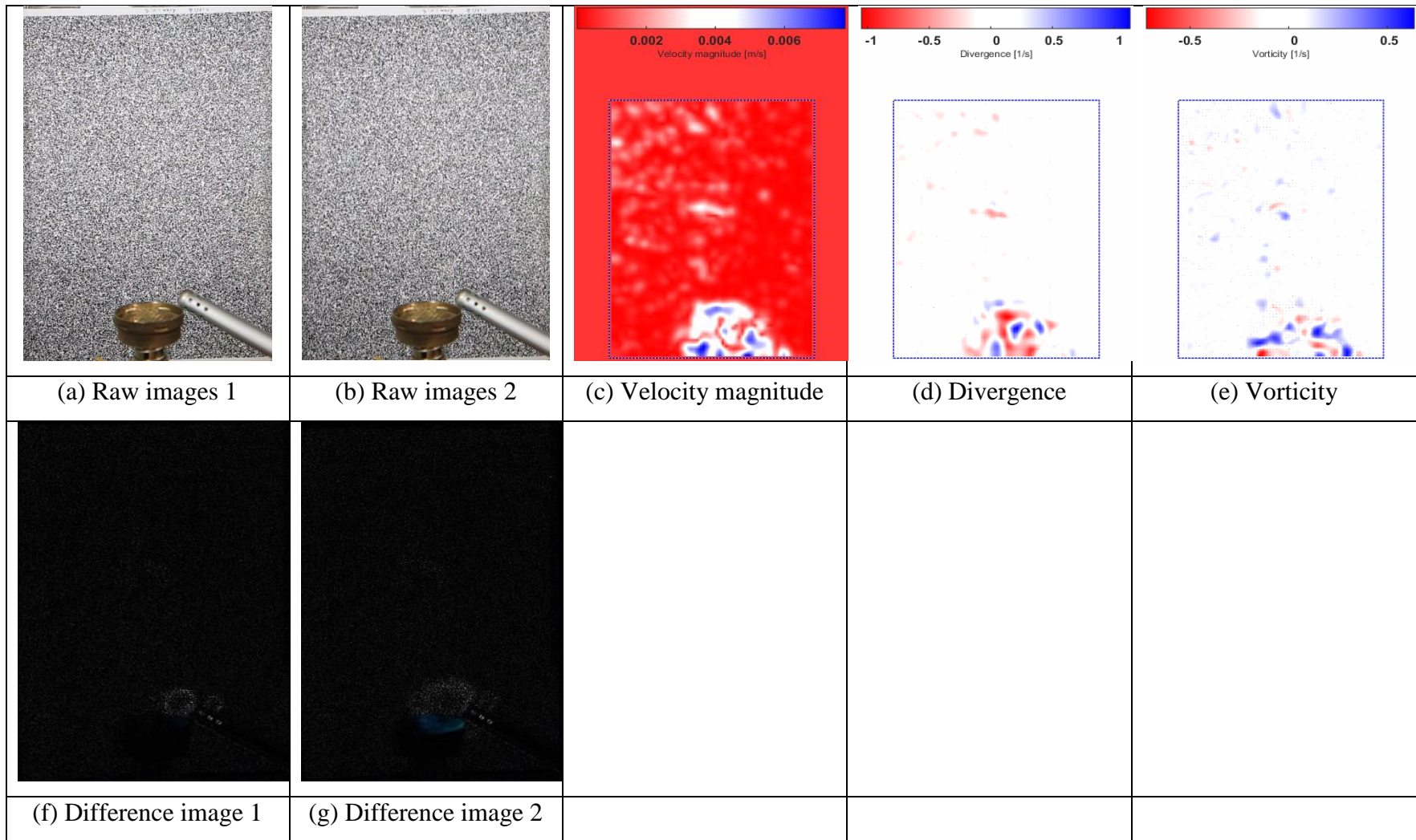


Figure 35: PIV analysis images 1 at reference time for 240 fps

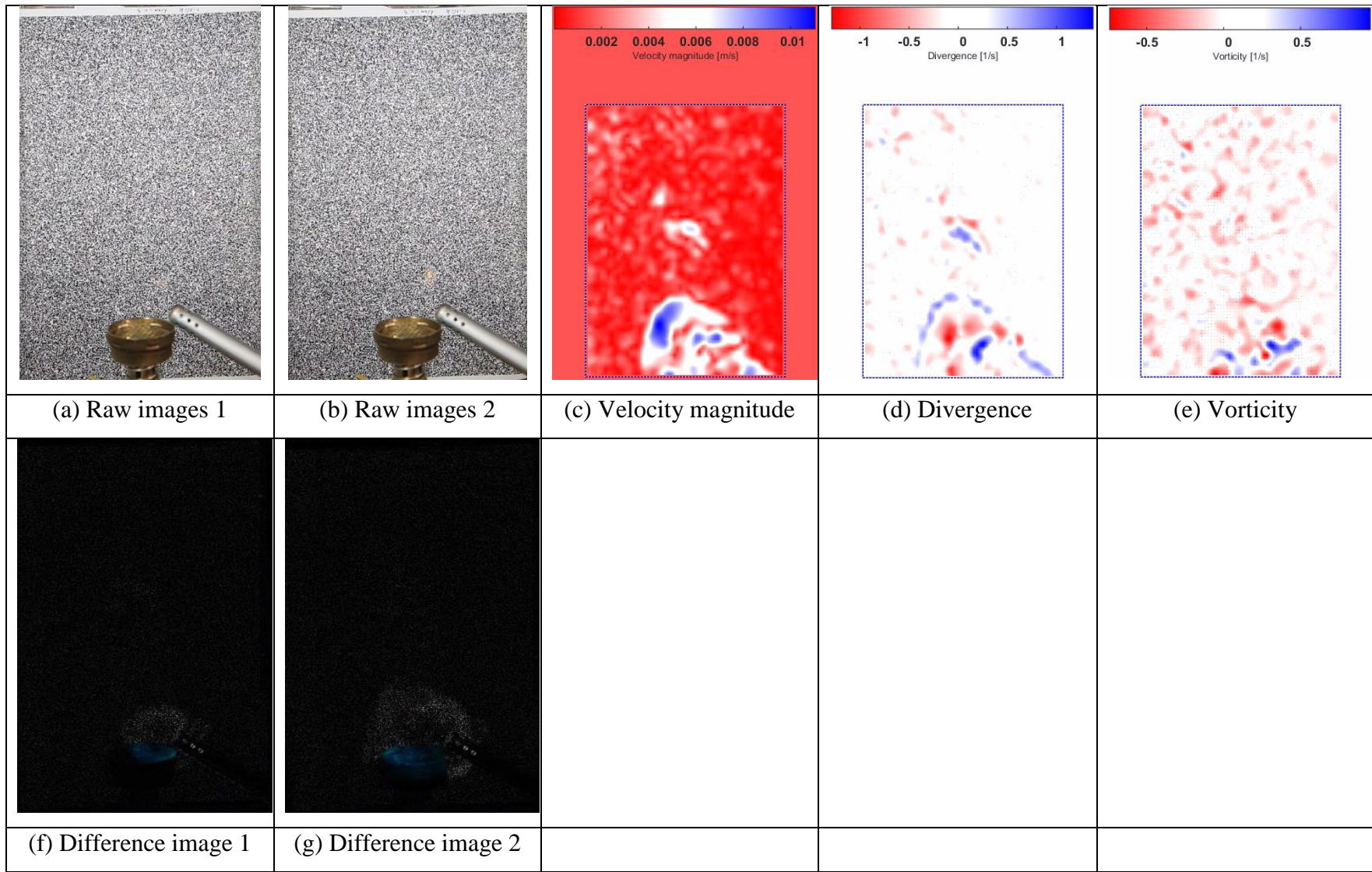


Figure 36: PIV analysis images 2 at reference time for 240 fps

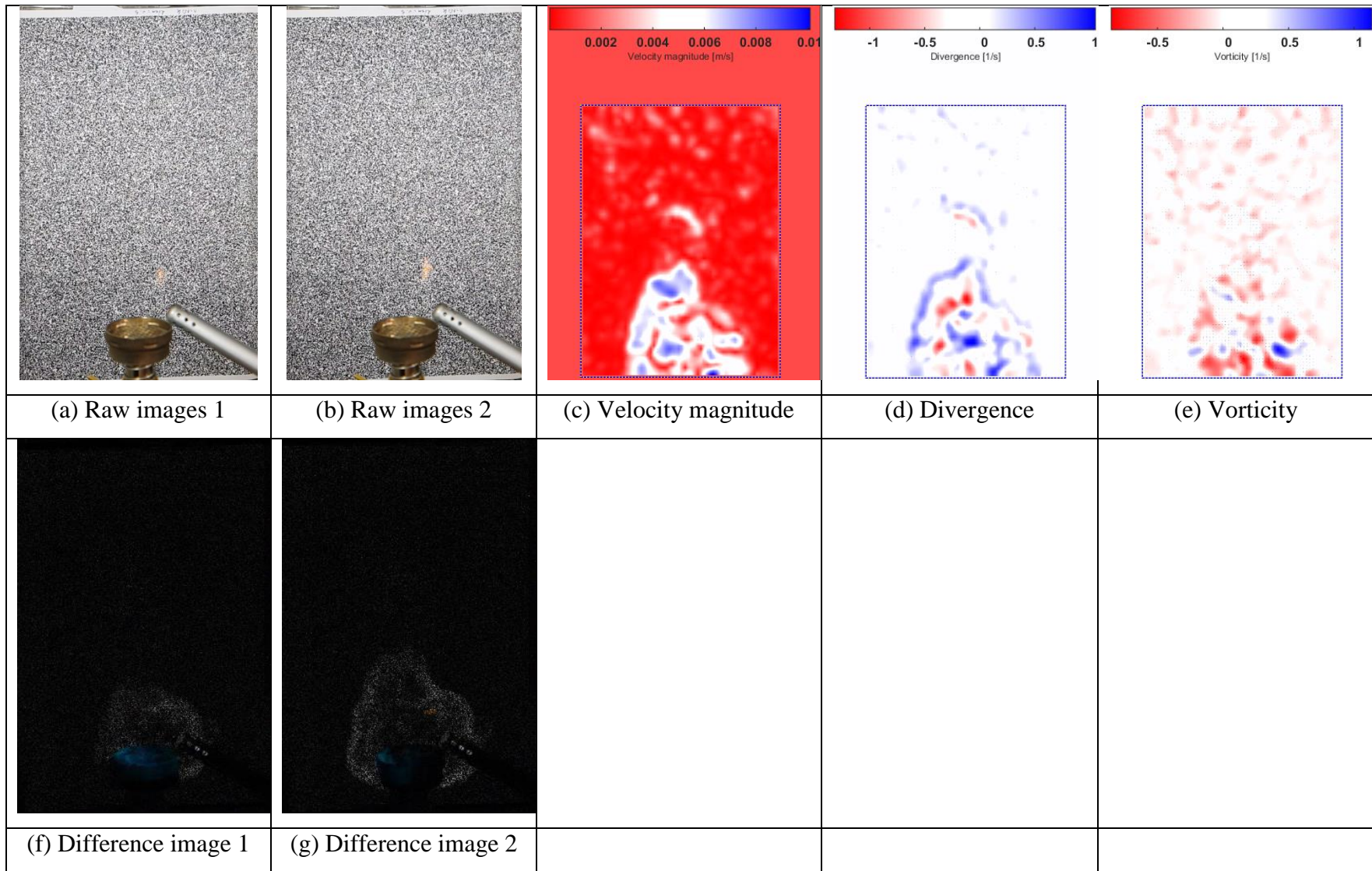


Figure 37: PIV analysis images 3 at reference time for 240 fps

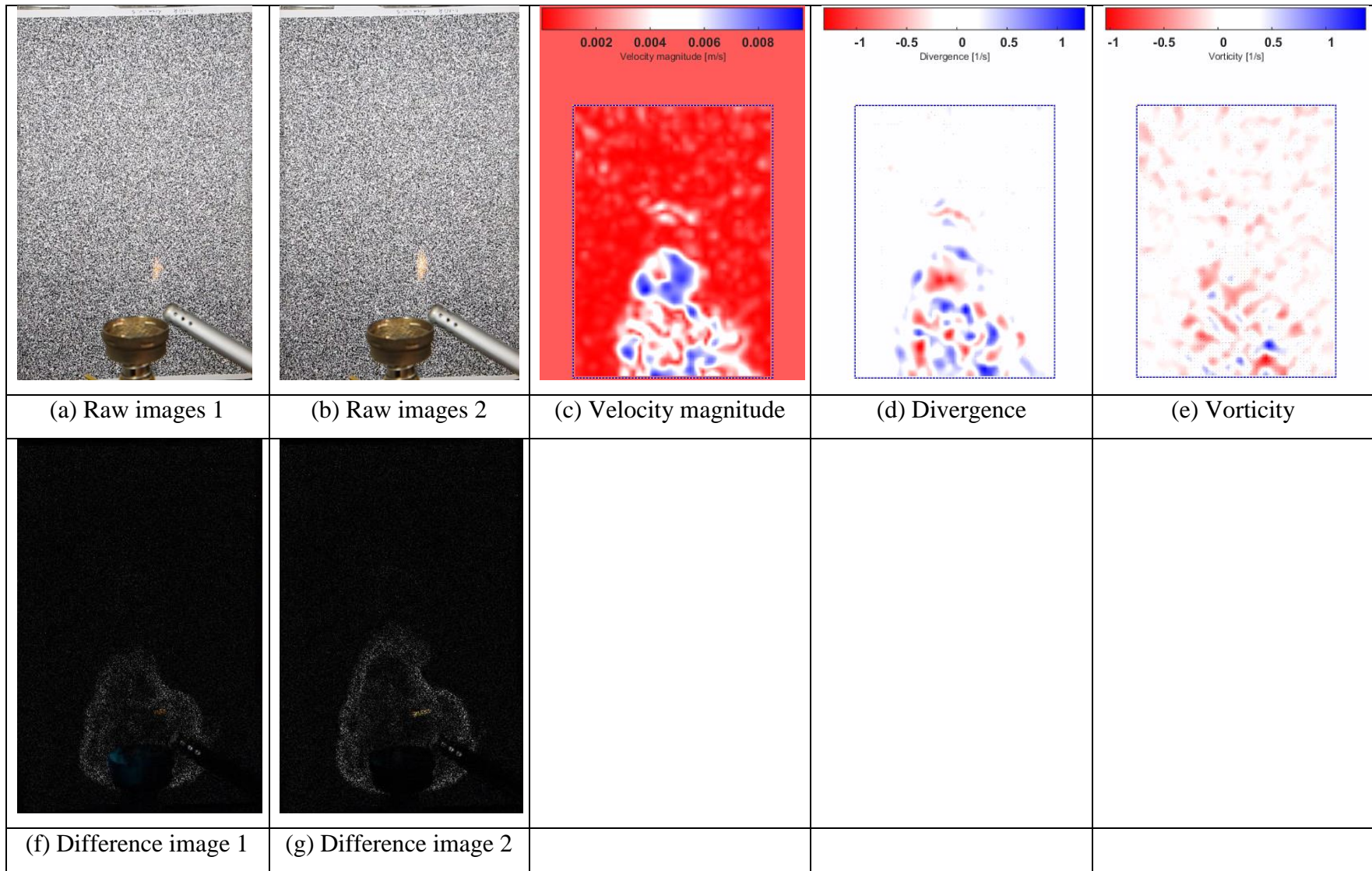
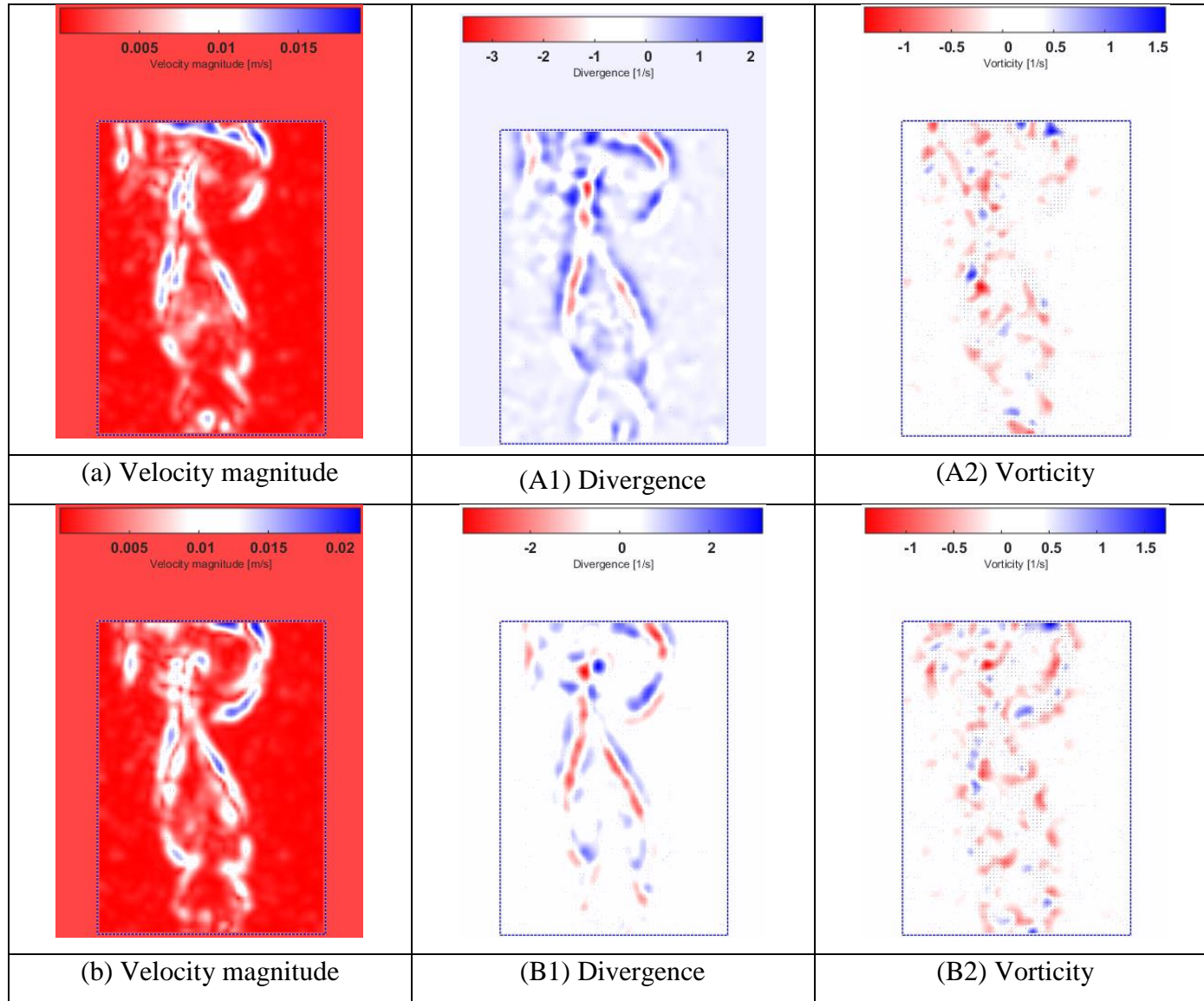


Figure 38: PIV analysis images 4 at reference time for 240 fps

After 5 sec from reference time:



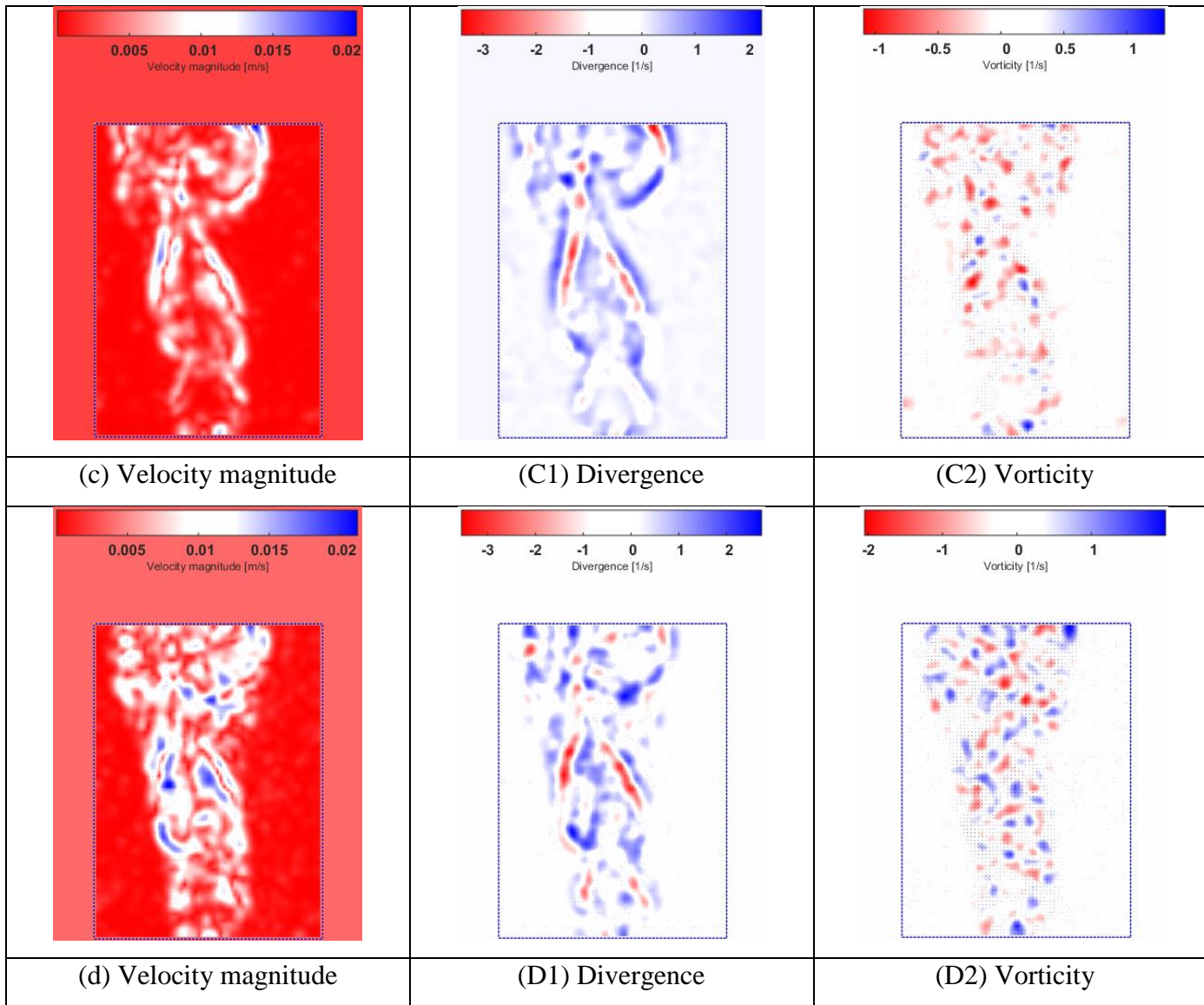
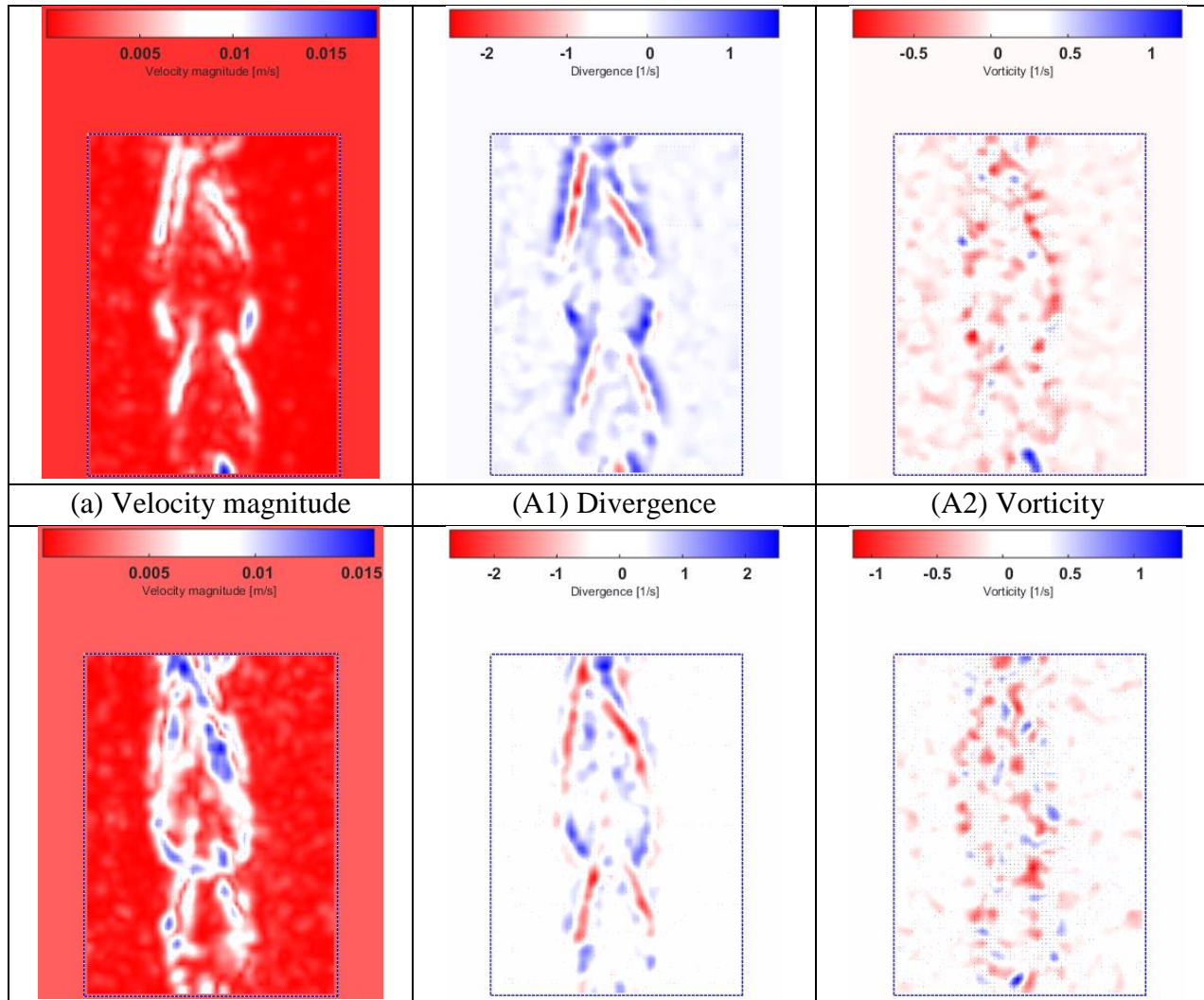


Figure 39: PIV analysis images after 5 sec for 240 fps

After 10 sec from reference time:



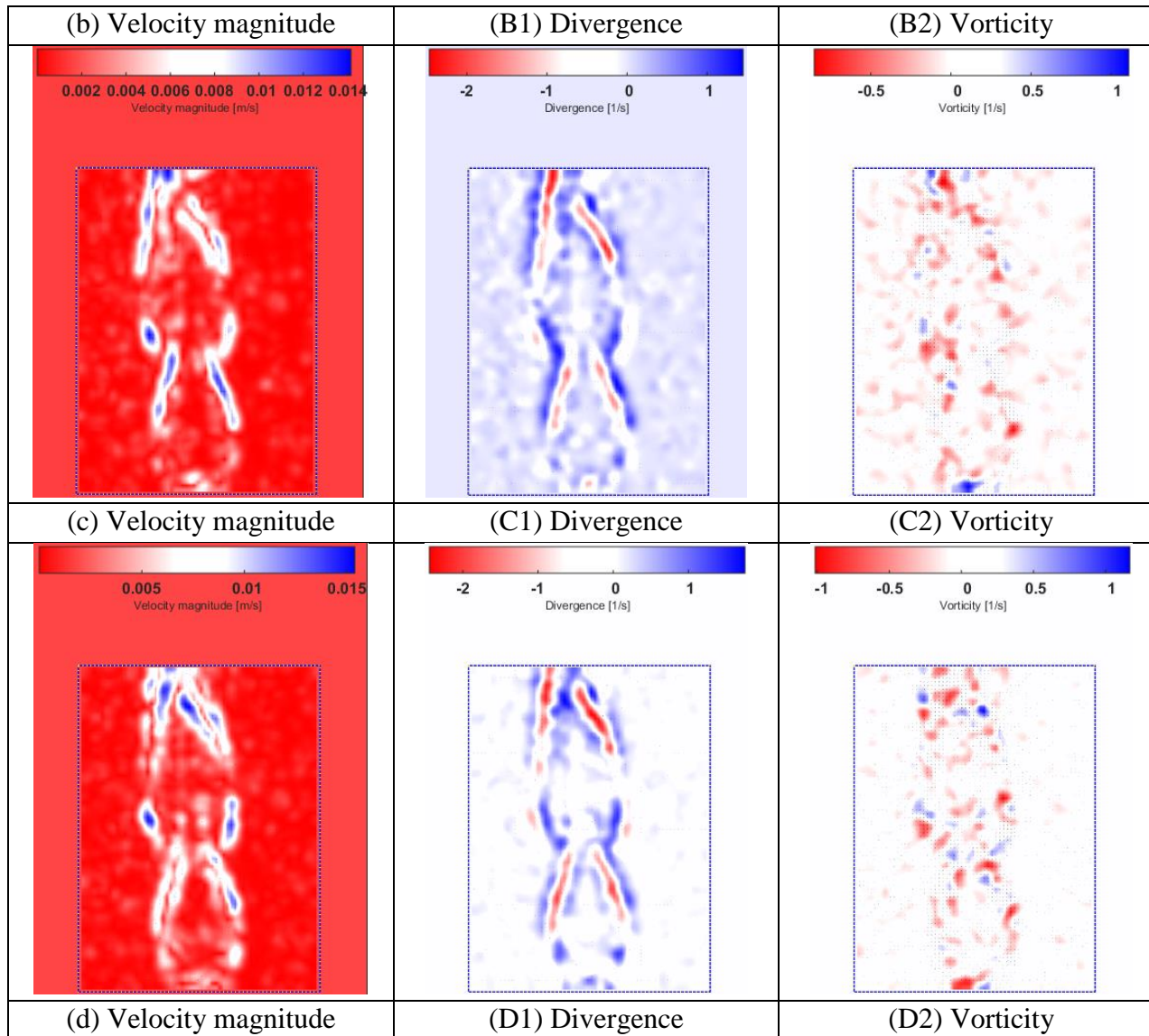


Figure 40: PIV analysis images after 10 sec for 240 fps

As stated above in the literature review, the structure of premixed flames was in conical shape. It could be observed from all the figures and specially from vorticity and absolute difference images (black and white) that “**flame structure is conical**” and results shows that premixed flames only has single flame structure/zone. Flame zones occur when more combustion takes place at the flame front as in the case of diffusion flame. Length of the flame can be observed by the percentage of air in the burner; more percentage of air is fed into burner, longer will be the structure of flame. Currently, inlets air and fuel were not controlled. Air and fuel into inlets should be more controlled and it will make it possible to study the effect of surrounding air i.e. by changing air percentage and level of combustion completeness.

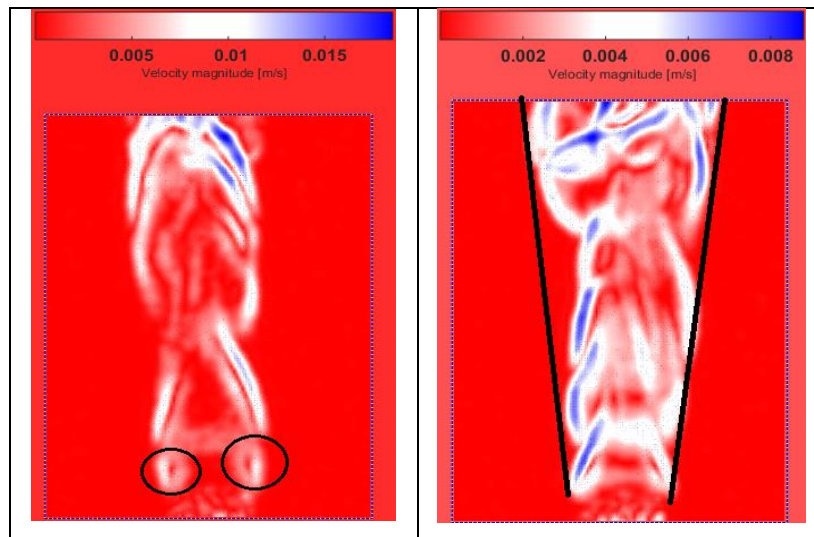


Figure 41: Effect of surrounding air and flame plume at stable position

Effect of surrounding air and flame plume can be observed in the Figure 41. From the figure above, it could be observed that flame propagate upwards in a axisymmetric way. The flame structure was diffused by surrounding air, due to the effect of temperature difference, as a jet which was coming in and starts to develop a vortex which was spreading out and moving upward. In few seconds after ignition, flame attains a stable structure. But this stability becomes irregular due to diffusion process. Flame displaced in the velocity range of 12 - 15 mm/sec. Air diffusion increases velocity and maximum velocity which has been observed in this experiment is 20 mm/sec.

As we increase number of frames per second (fps), we can get more information about motion (i.e. more images per second), but down side of having high fps is lighting requirement will increase significantly. There are also limitation to CMOS (Complementary metal-oxide semiconductor) based camera sensor to handle capture and shutter, because CMOS sensor works like a scanner. It scans down the image, which is called rolling shutter (figure 42). If we need to really on 240 fps or higher frames per seconds, then we need to compensate for resolution, varying background and shutter speed.



Figure 42: Effect of rolling shutter on moving propeller [37]

4.4. BOS experiment using mirror:

The PIV images are two-dimensional (2-D); therefore it is not possible to get 3-D images. This study is not focused on 3-D images but concerns with multiple side-views using a mirror because the cameras/lens are an expensive part of the experiment. Another experiment is done by adding a mirror in the existing setup to get a different view (figure 43). The addition of the mirror is to check the applicability of a mirror used in BOS experiment, or in other words “is it possible to get results/images using a mirror”?

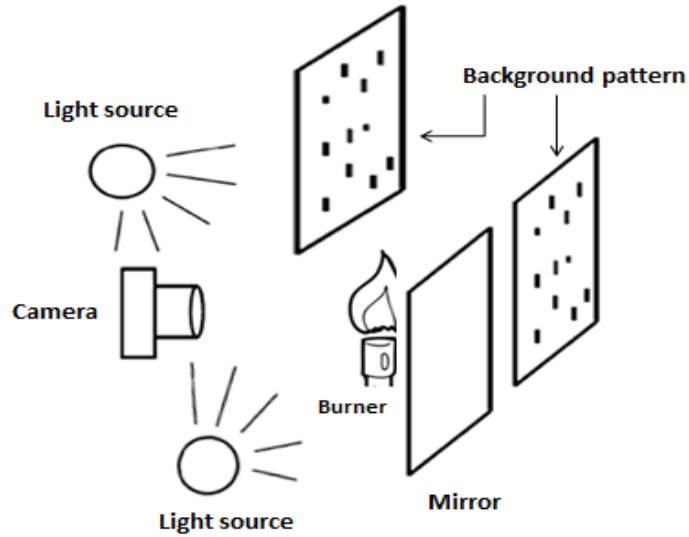


Figure 43: Diagram depicting experimental setup using mirror

4.4.1. 30 frame per second:

Start ignition $t = 0$ (taken as reference for time):

Following are figures shown in sequence, first at $t = 0$ then after 5 and 10 sec of ignition to show progress of flame flow from burner base to top at 30 fps and flame structure.

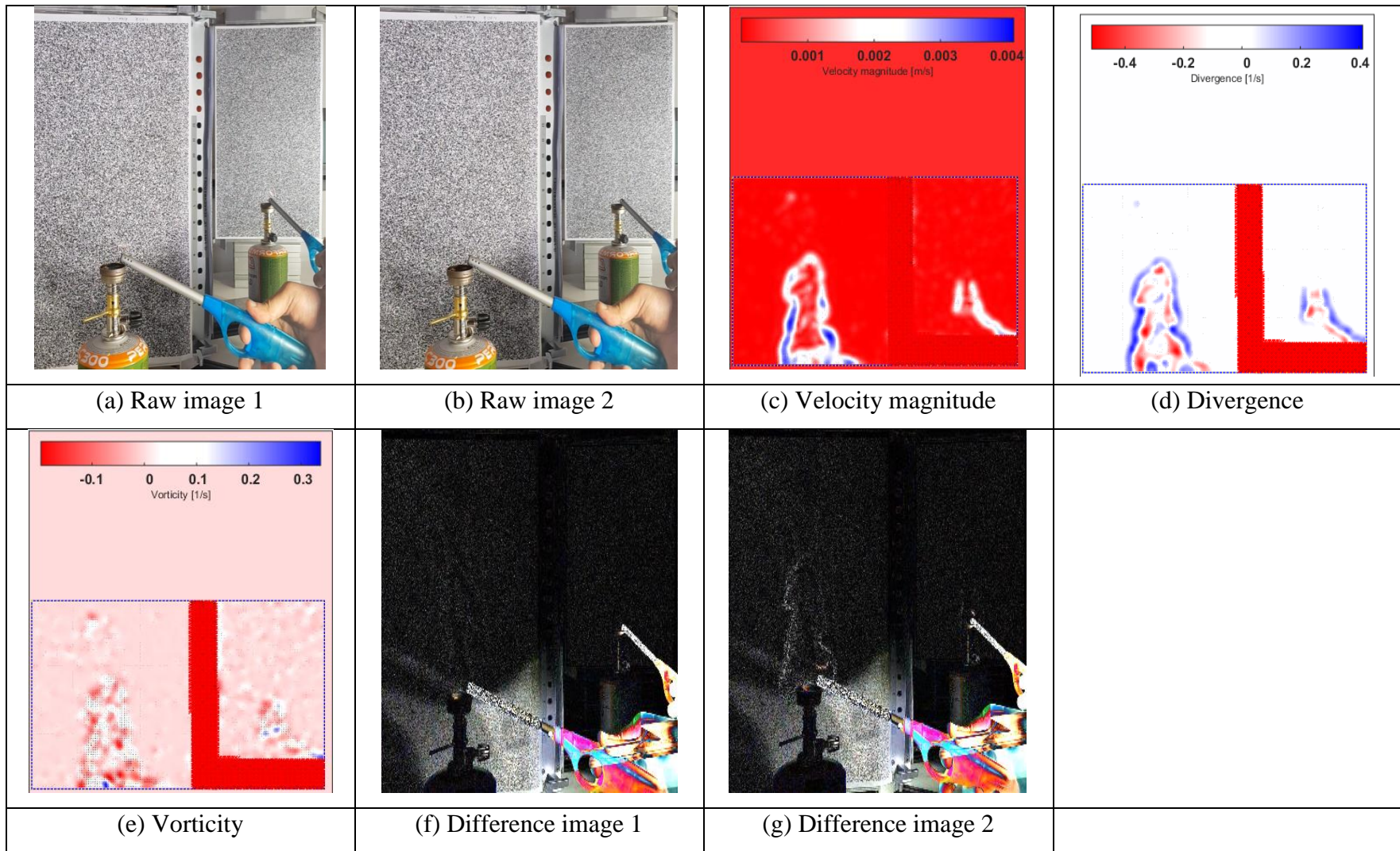


Figure 44: PIV analysis images 1 using mirror at reference time for 30 fps

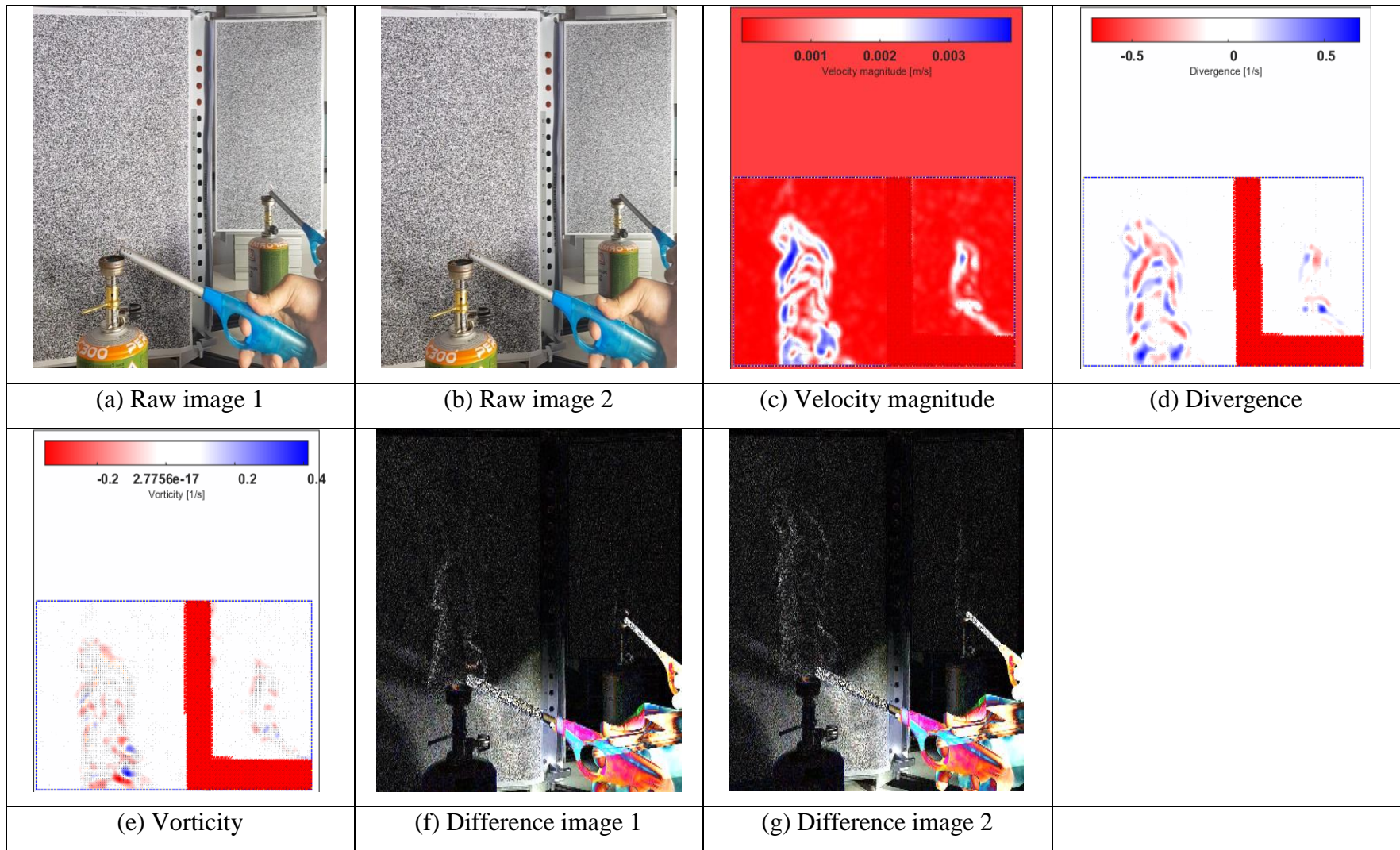


Figure 45: PIV analysis images 2 using mirror at reference time for 30 fps

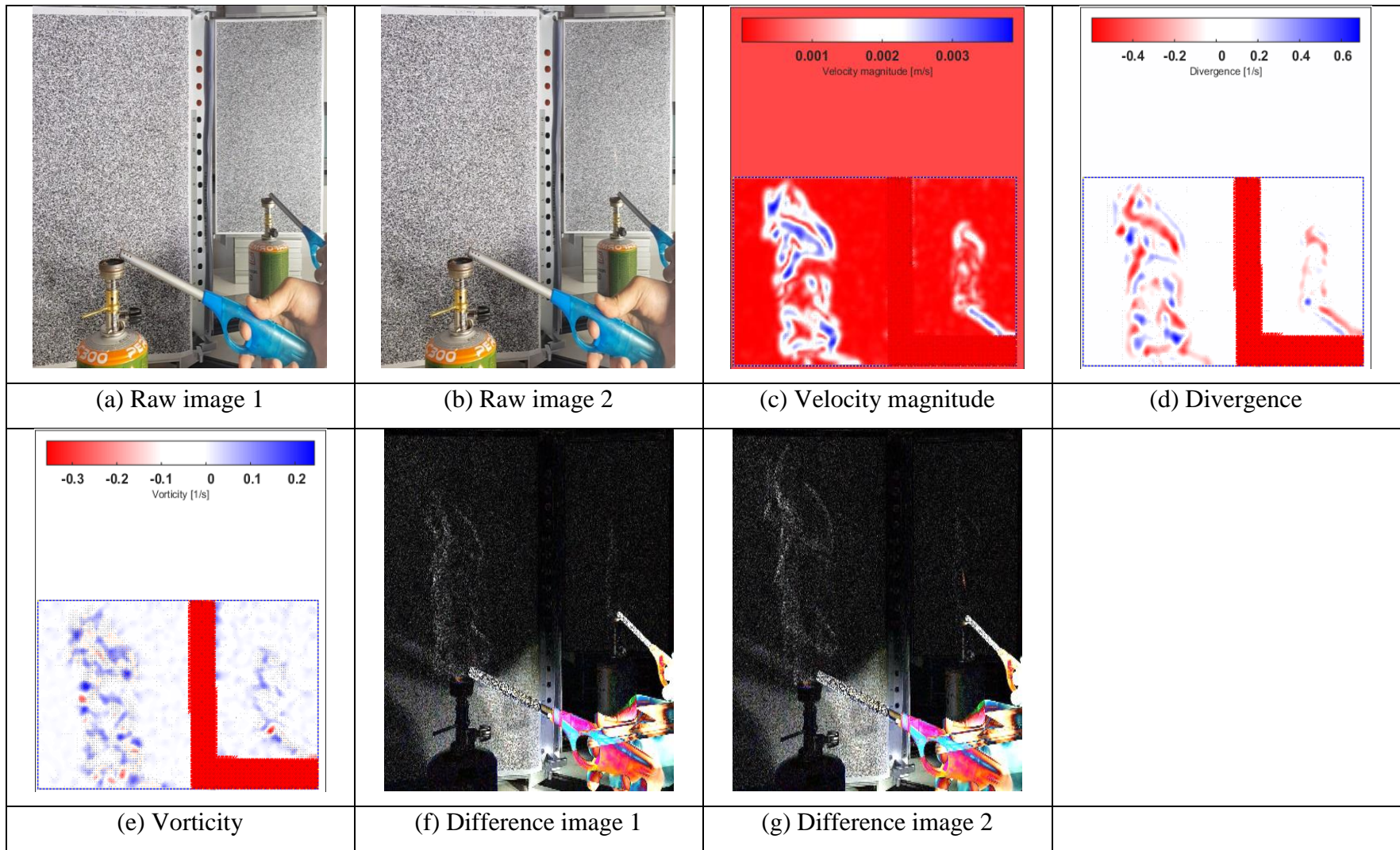
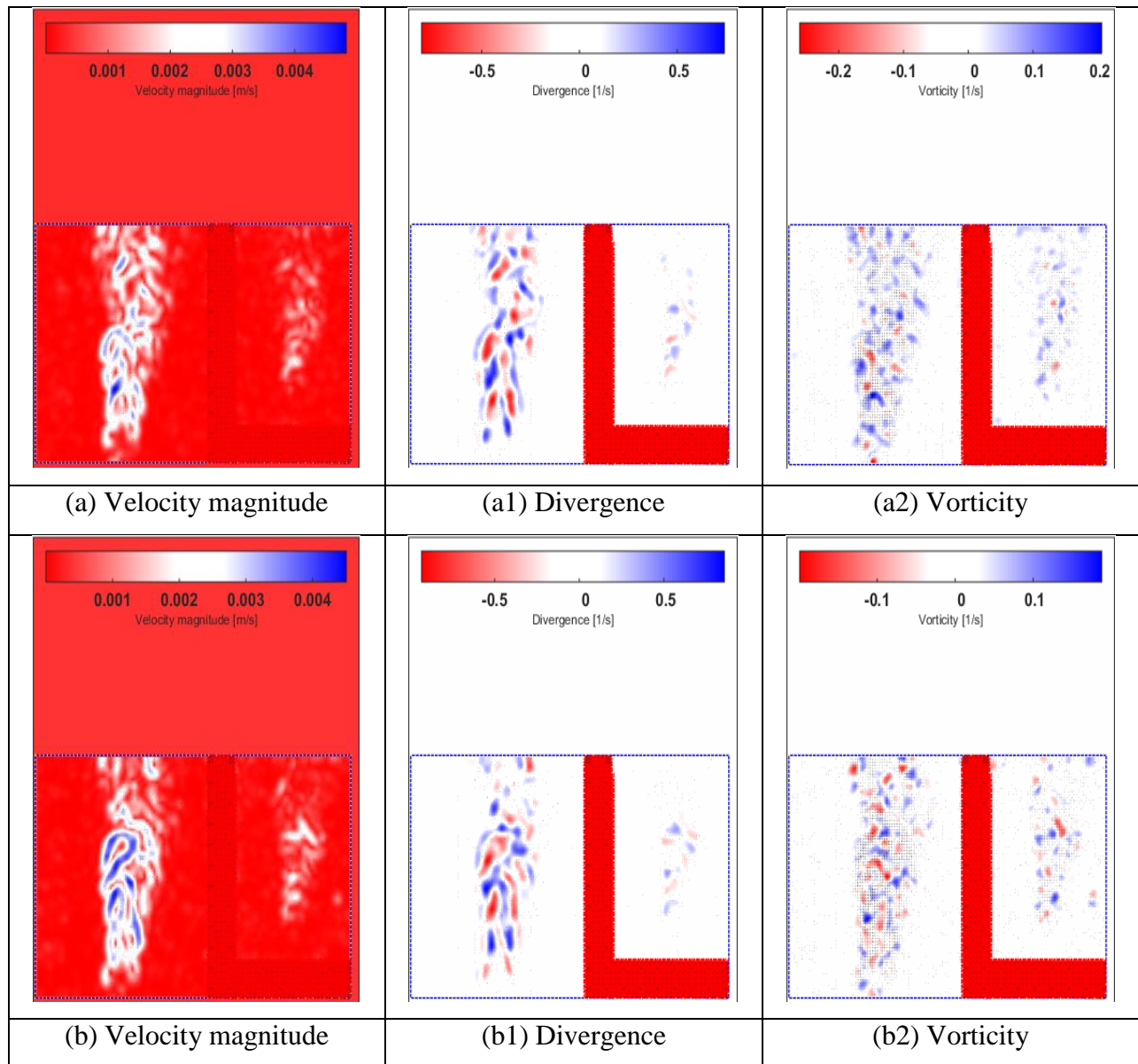


Figure 46: PIV analysis images 3 using mirror at reference time for 30 fps

After 5 sec from reference time:



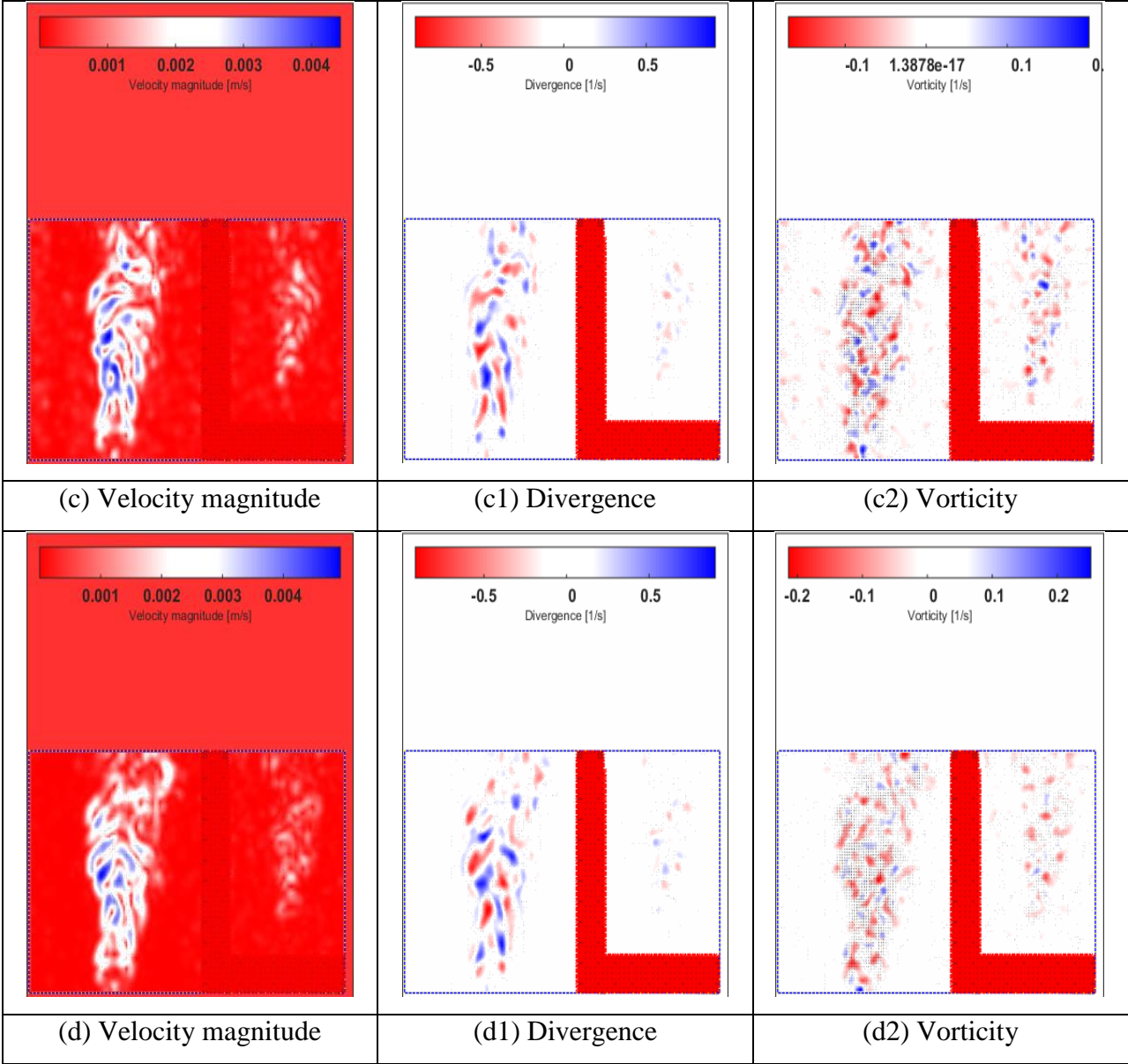
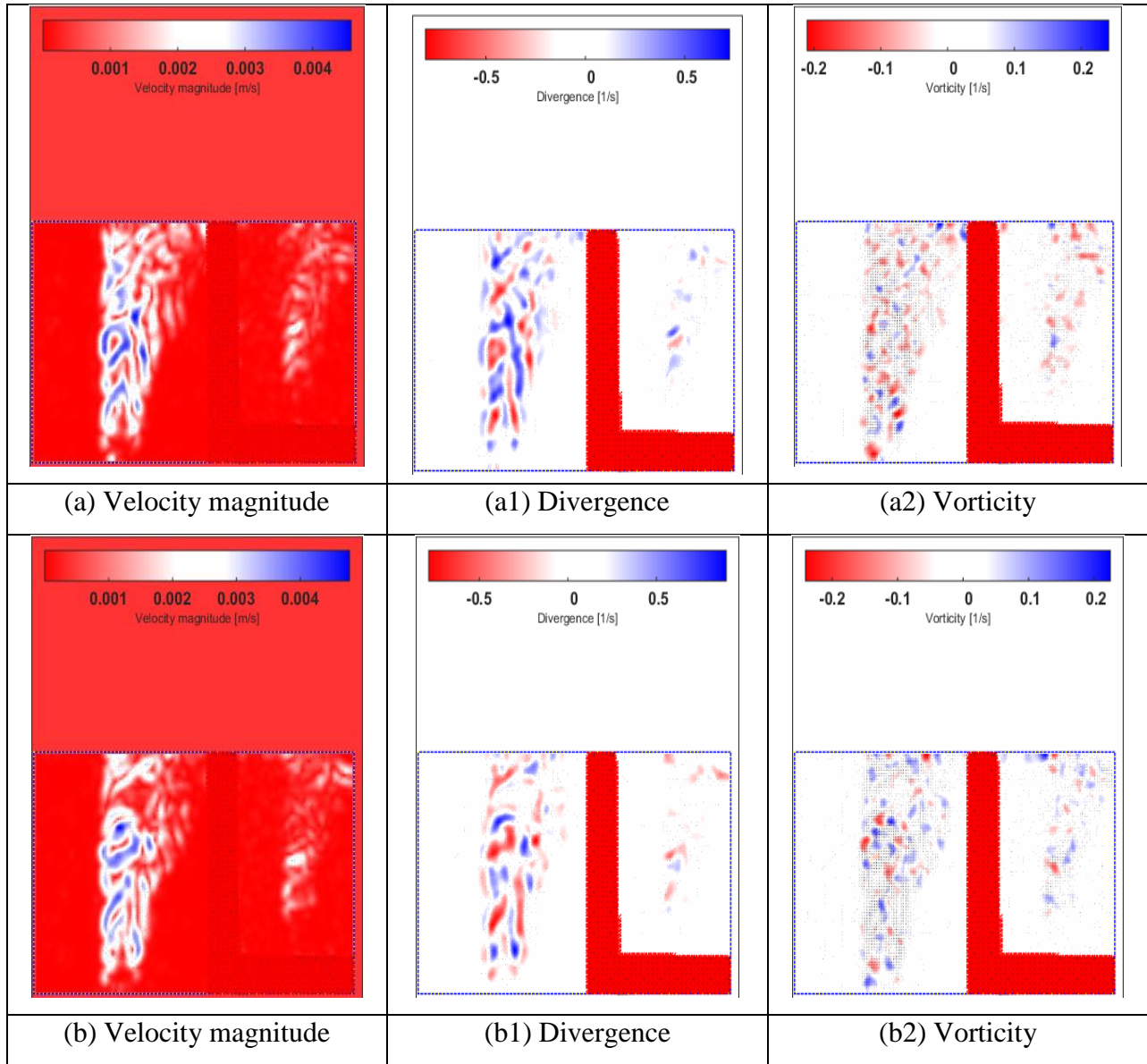


Figure 47: PIV analysis images using mirror after 5 sec for 30 fps

After 10 sec from reference time:



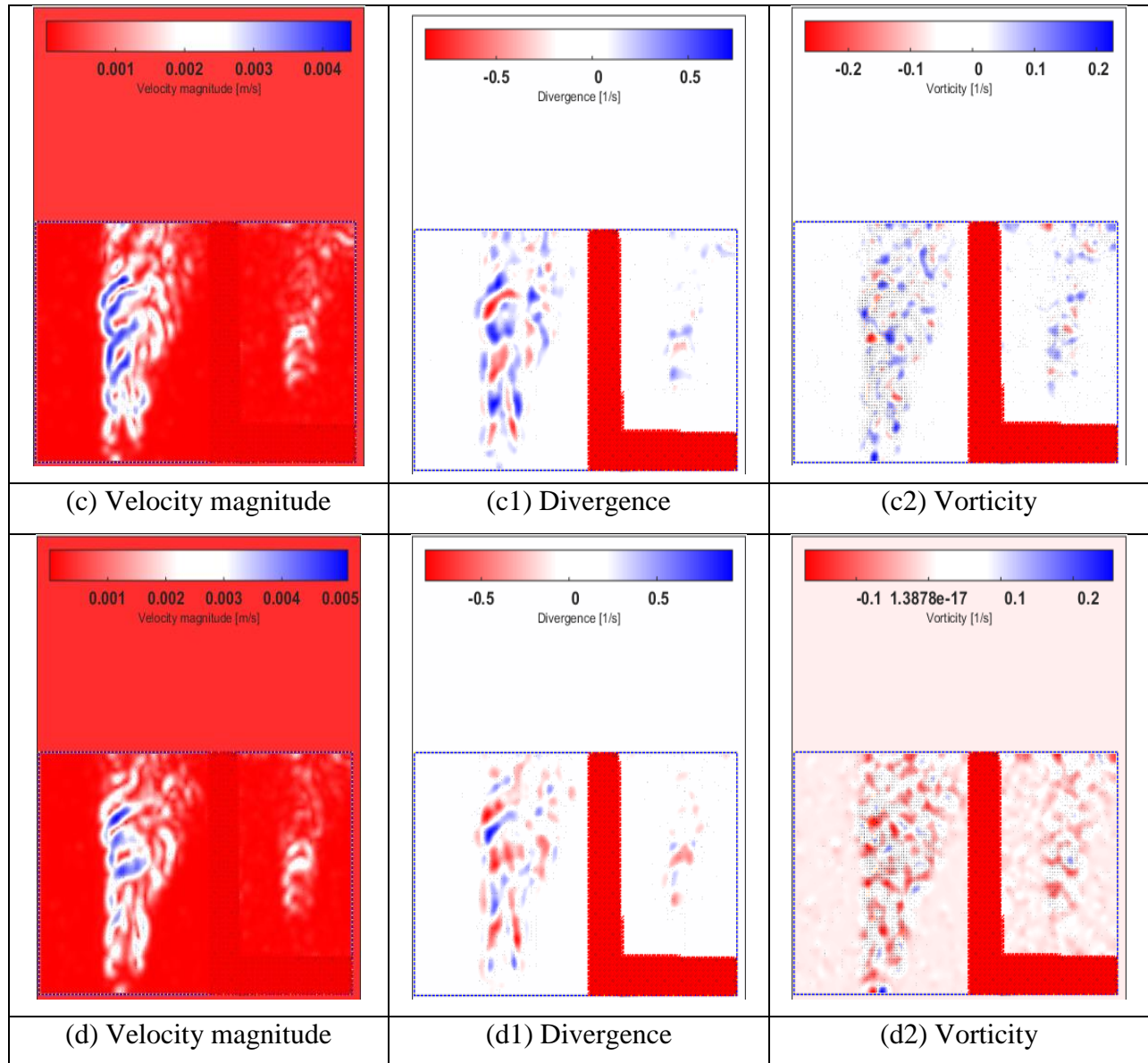


Figure 48: PIV analysis images using mirror after 10 sec for 30 fps

4.4.2. 60 frames per second:

Start ignition $t = 0$ (taken as reference for time):

Following are figures shown in sequence, first at $t = 0$ then after 5 and 10 sec of ignition to demonstration movement of flame flow from bottom to top at 60 fps and flame structure.

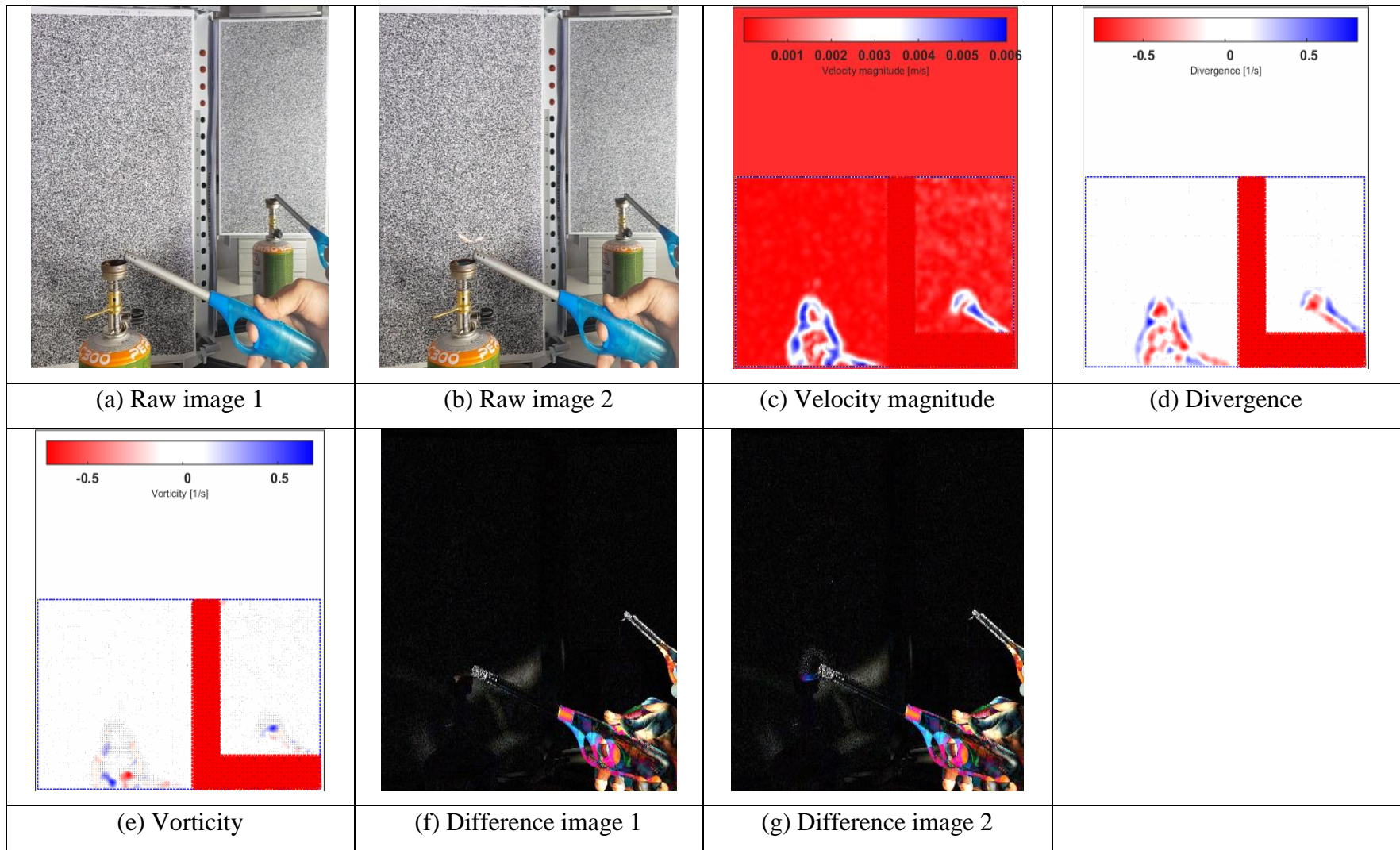


Figure 49: PIV analysis images 1 using mirror at reference time for 60 fps

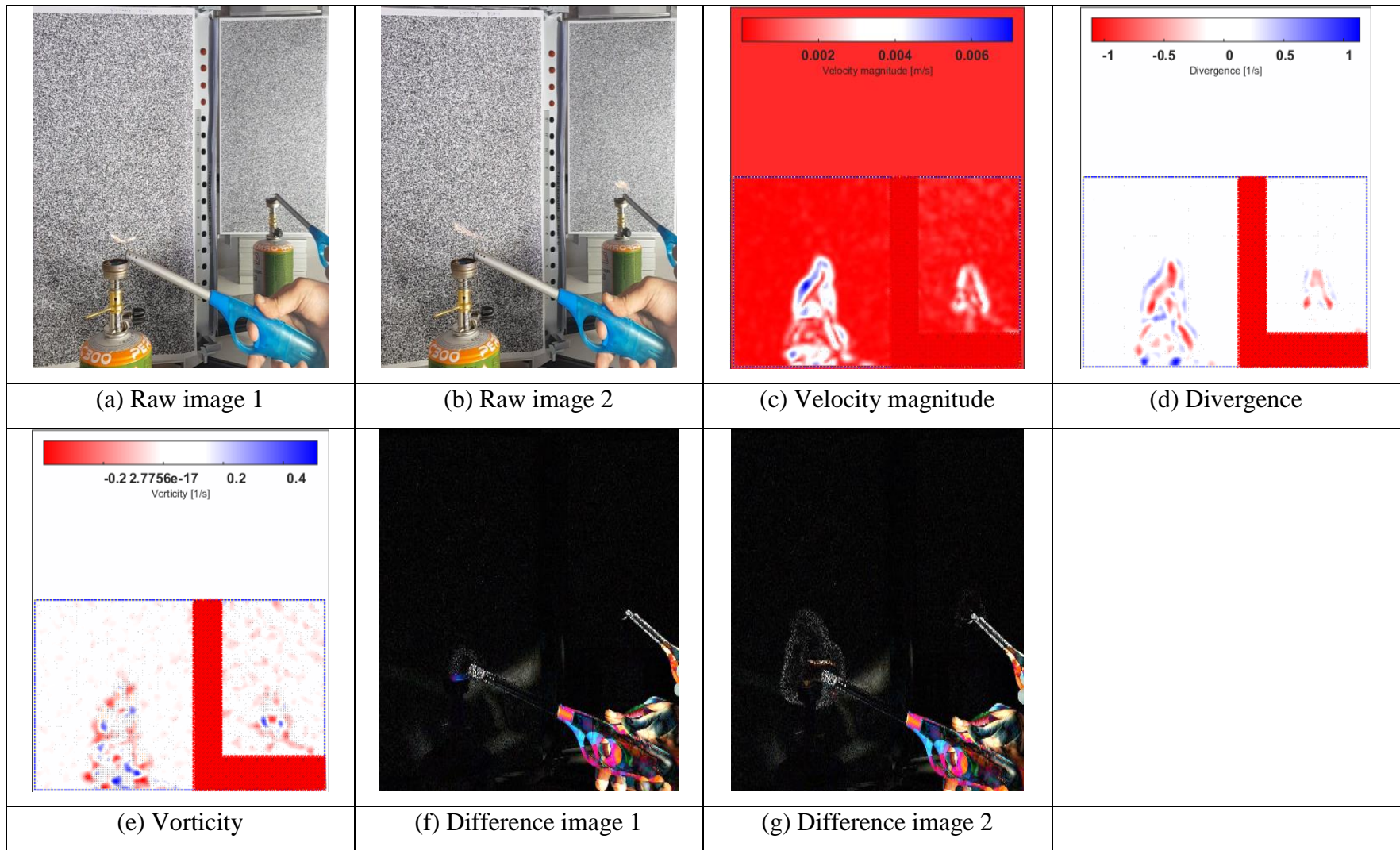


Figure 50: PIV analysis images 2 using mirror at reference time for 60 fps

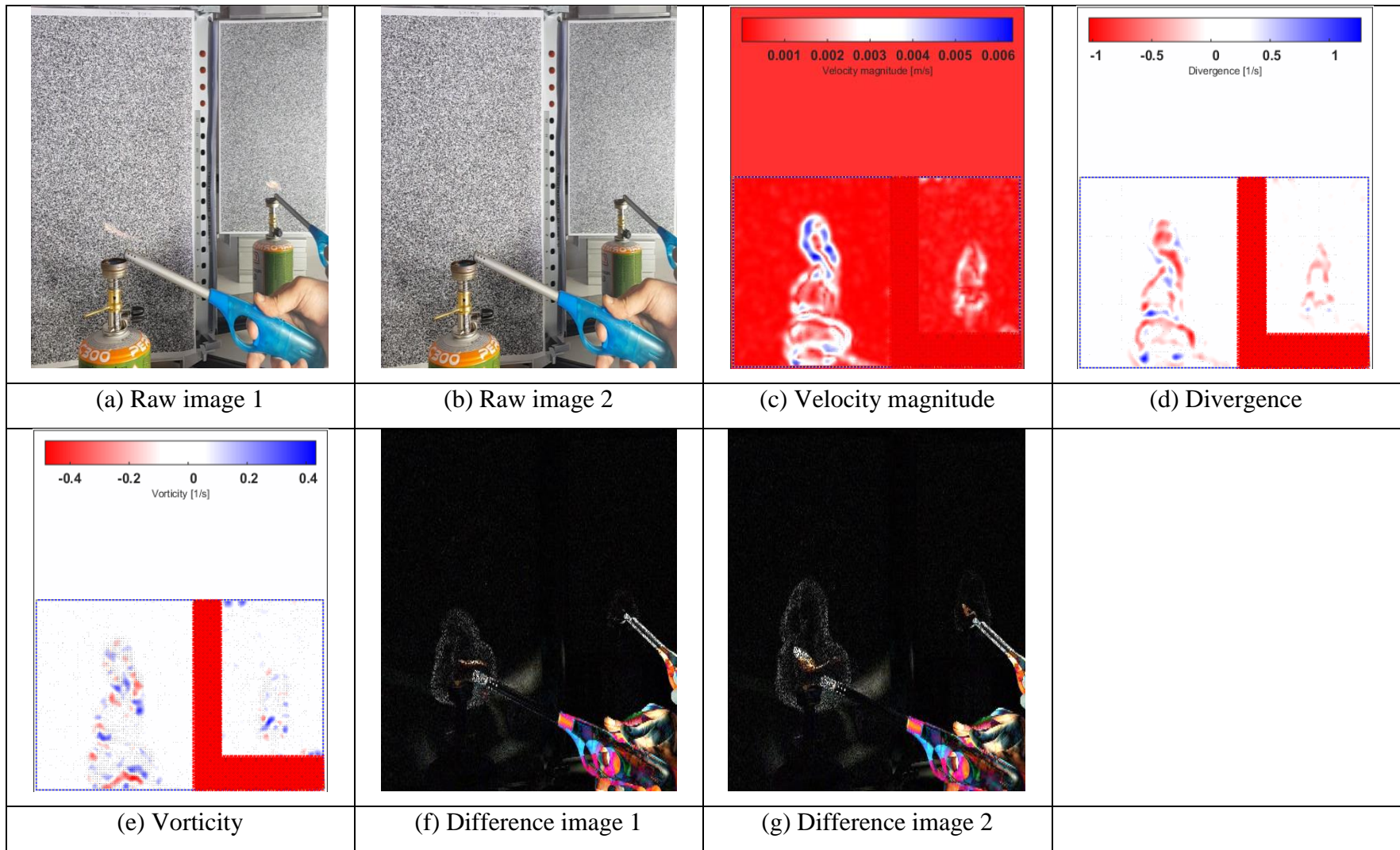


Figure 51: PIV analysis images 3 using mirror at reference time for 60 fps

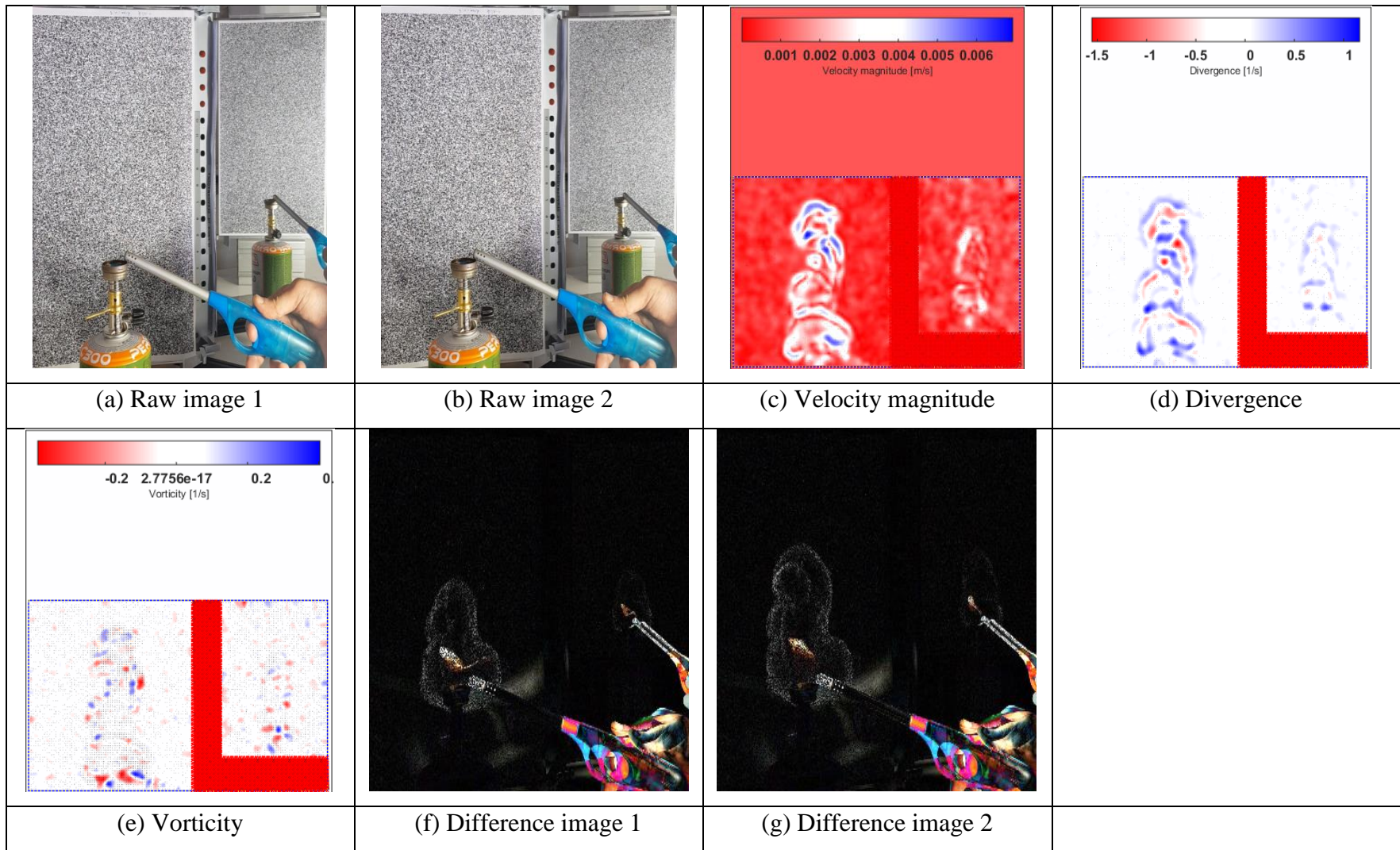
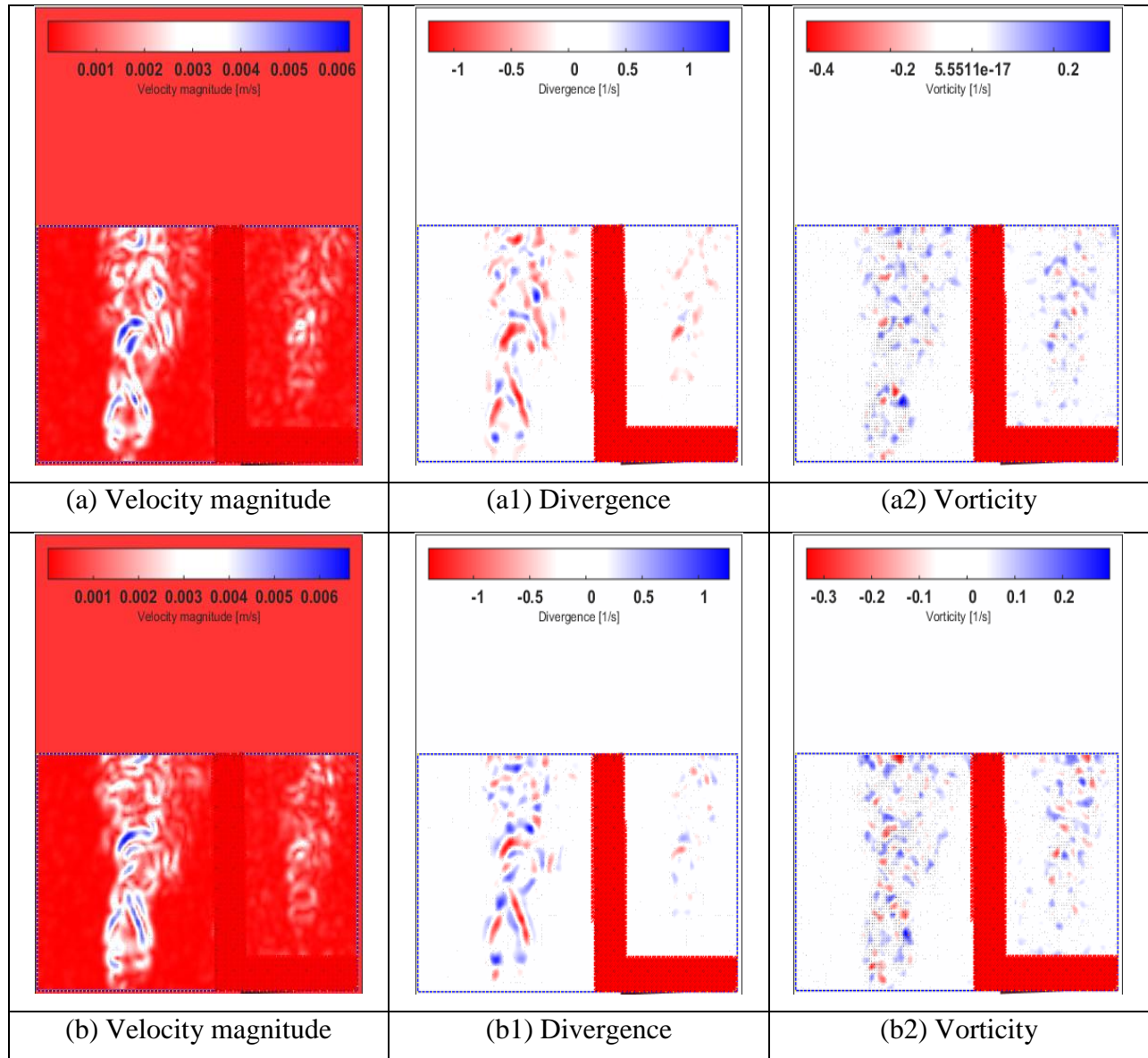


Figure 52: PIV analysis images 4 using mirror at reference time for 60 fps

After 5 sec from reference time:



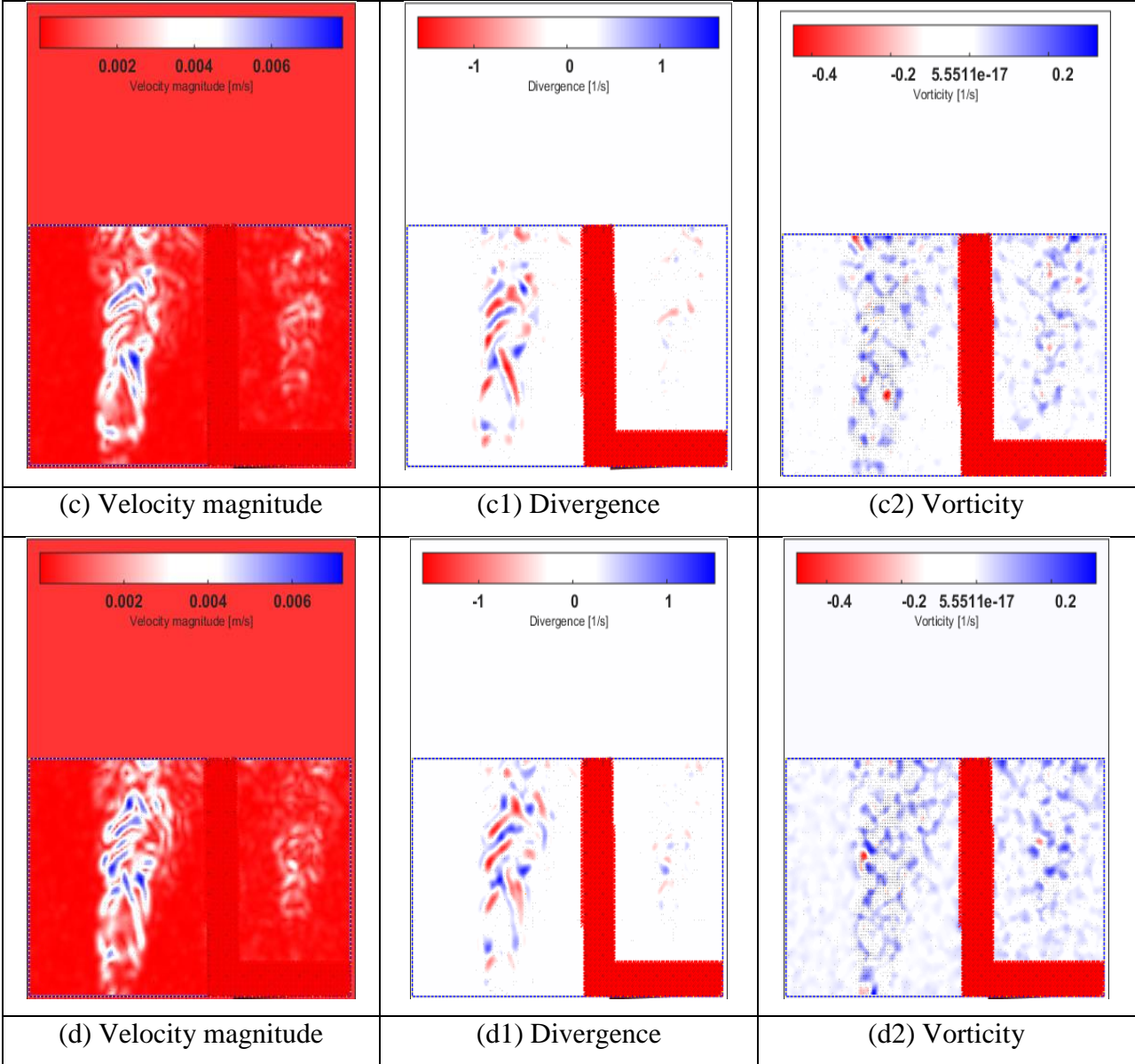
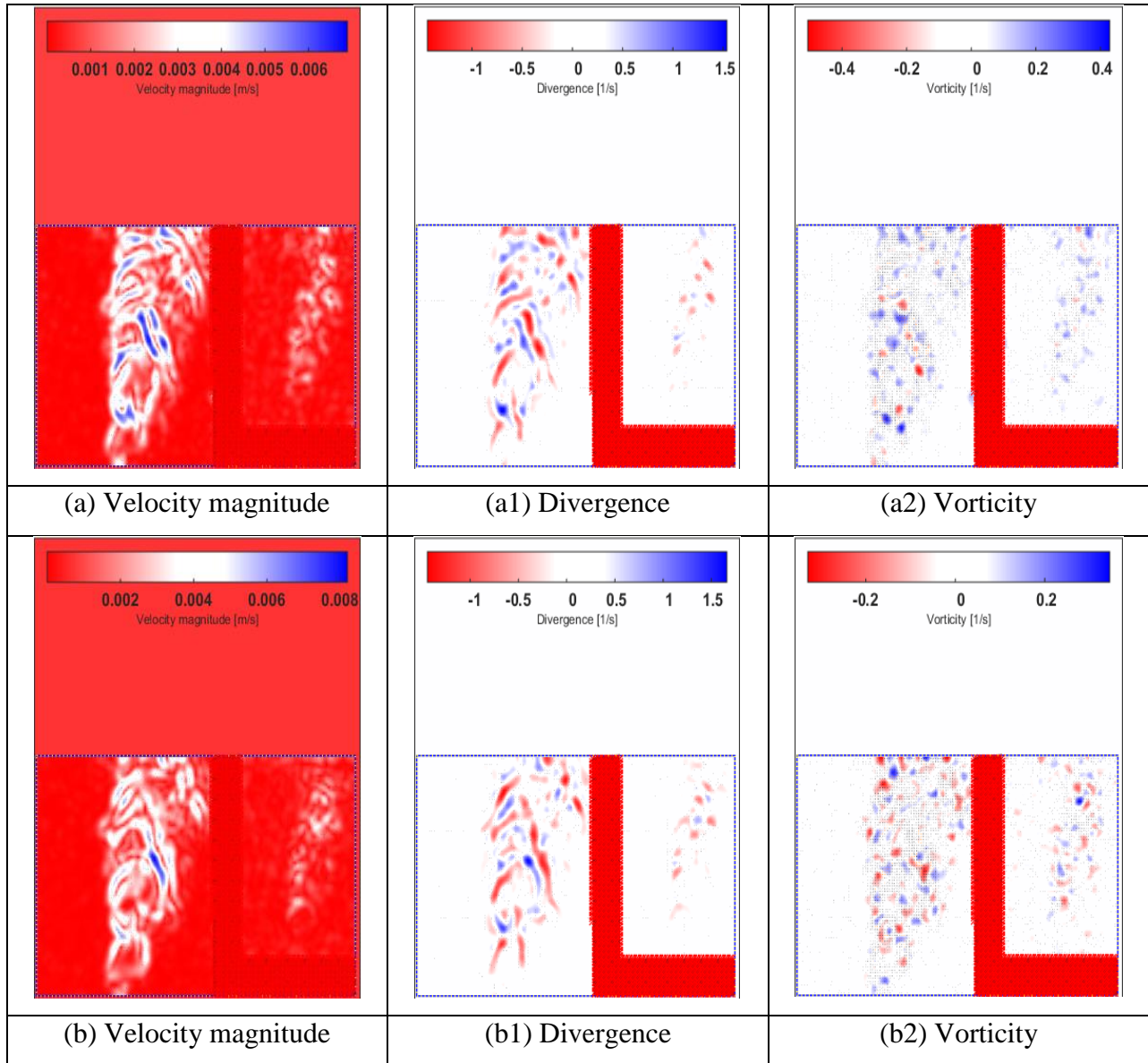


Figure 53: PIV analysis images using mirror after 5 sec for 60 fps

After 10 sec from reference time:



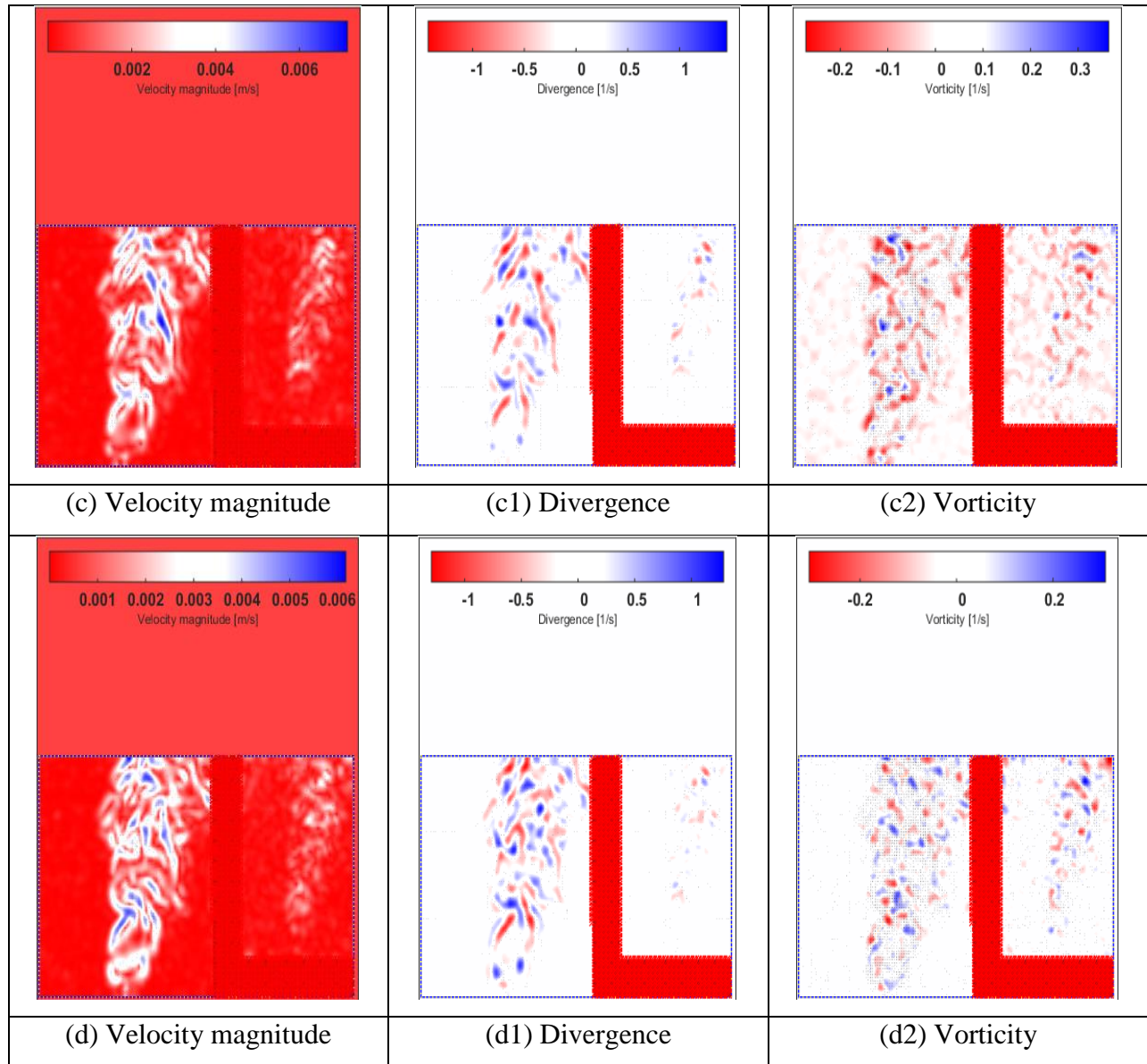


Figure 54: PIV analysis images using mirror after 10 sec for 60 fps

Another experiment was done to get different view of flame propagation using mirror. Some of the observation are listed below:

- Very light flame propagation through mirror could be observed but we had lost most of intensity/energy between mirror image and normal image. One of the reason behind it is the internal reflection of the mirror. Light goes into glass, some of the light get reflected back to viewer but some part shifted slightly and get into transparent material of mirror. When light hit coaster layer reflected back, due to internal reflection we get two images.

- As focus was on two backgrounds (one normal and other 90 degree flipped due to mirror), so it is difficult to attain sharp focus on both backgrounds and also on subject.
- A better first surface mirror is required to prevent internal reflection or loss of light intensity and to get sharp focus on schlieren object with backgrounds.

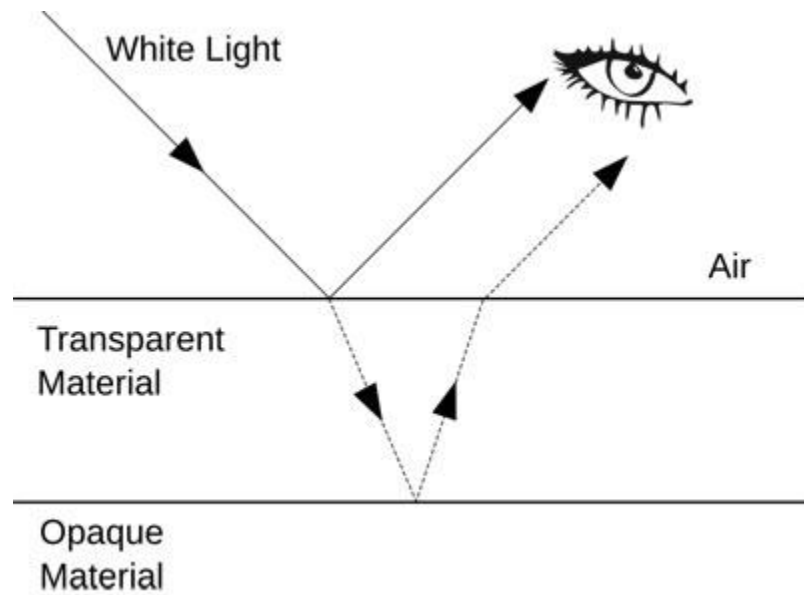


Figure 55: Internal reflection in mirror [38]

Chapter 5: Conclusion & Future Work

5.1. Conclusion:

In the present study, an experimental setup for studying premixed flames has been developed and used for tests. The focus has been to study structure and propagation of flame. The images of flame was acquired by using smartphone camera and the velocities of flame front computed using PIVlab (MATLAB application) to find out how fast flame front propagates from bottom to top. Moreover, a MATLAB program was developed which can subtract two images to demonstrate flame structure.

The available camera frame rate is the limiting factor for smartphone background oriented schlieren velocimetry. It should be sufficiently high that the imaged flow experiences a few pixels displacement between frames. If fps (frame per second) rate increases, then more light is need on the background as camera have less time to take an image (more frames in a second). The camera uses CMOS sensor which works like a scanner, it scans down the chip to create image. Down side of using CMOS sensor, if an object is moving in the same direction of scanner (normally top to bottom) then sensor can stretched out the shape of the object. Inversely if the object is moving in opposite direction, then it is going to compress that object.

Real flows are compressible, 3-dimensional, and unsteady in nature. The 3D info of flow can be quantified by capturing data from multiple directions on the flow field. Effort was done to get information from two different sides, one normal camera plane and the other 90 degree flipped using mirror that was focus on another BOS background. Main purpose of using mirror is to capture flame images from a side view, but some issues like loss of light intensity, internal reflection (first surface mirror) needs to be addressed.

5.2. Future work:

At the end of this report, it is viable to point-out some potential future research areas. Following are some workable future research areas:

- This method can be used to study effect of air percentage (if air inlet is controlled), burner diameter and different fuel types on structure of flame.
- This technique can be used for gas leakage detection, gases which are colourless and odourless.
- It is highly recommended to simulate premixed flame using CFD (Computational Fluid Dynamic) modeling to compare the experimental results.
- Use of natural background to measure density variation at large scale.

References

- [1] Roy, D. (1997). The cost of fires: a review of the information available, Home Office Publication Unit, London.
- [2] Drysdale, D. (1999). An introduction to fire dynamics, 2nd edition, pp. 1-130. John Wiley & Sons.
- [3] Warnatz, J., Mass, U., Dibble, R.W. (2006). Combustion: physical and chemical fundamentals, modelling and simulation, experiments, pollutant formation, 4th edition, Springer-Verlag Berlin Heidelberg.
- [4] Sevault, A. (2012). Investigation of the Structure of Oxy-Fuel Flames Using Raman Laser Diagnostics. Doctoral Theses, NTNU, <https://brage.bibsys.no/xmlui/handle/11250/235022> [Accessed 2 Jan. 2019].
- [5] YouTube. (2019). Premixed Flame vs Diffusion Flame - Fire Protection Engineering (FPE) Teaching Tool. [Online] Available at: https://www.youtube.com/watch?v=_gqO3ncLfRg [Accessed 30 May 2019].
- [6] Hottel, H.C., and Hawthorne, W.R. (1949). *3rd Symposium (International) on Combustion*, pp. 255-266. Williams and Wilkins, Baltimore.
- [7] Wang, Q. (2012). *Advanced Optical and 3D Reconstruction Diagnostics for Combustion and Fluids Research*, pp. 30-55, PhD thesis, University of Sheffield.
- [8] Richard, H., Raffel, M., Rein, M., Kompenhans, J. and Meier, G. (2002). Demonstration of the applicability of a Background Oriented Schlieren (BOS) method, *Laser Techniques for Fluid Mechanics*, pp.145-156.
- [9] Tokgoz, S., R. Geisler, L. J. A. van Bokhoven and B. Wieneke (2012). Temperature and velocity measurements in a fluid layer using background-oriented schlieren and PIV methods, *Measurement Science and Technology* **23**(11): 115302.
- [10] Richard, H., Raffel, M. (2001). Principle and applications of the background oriented schlieren (BOS) method, *Measurement Science and Technology* **12** 1576–85.
- [11] Meier, G.E.A. (2002). Computerized background-oriented schlieren, *Exp. Fluids* **33** 181–7.

- [12] Dalziel, S.B., Hughes, G.O., Sutherland, B.R. (2000). Whole-field density measurements by 'synthetic schlieren', *Exp. Fluids* **28** 322–35.
- [13] Venkatakrisnan, L., Meier, G.E.A. (2004). Density measurements using the background oriented schlieren technique, *Exp. Fluids* **37** 237–47.
- [14] Merzkirch W., Egami Y. (2007). Density-Based Techniques. In: Tropea C., Yarin A.L., Foss J.F. (eds) *Springer Handbook of Experimental Fluid Mechanics*. Springer Handbooks. Springer, Berlin, Heidelberg.
- [15] Elsinga, G.E., van Oudheusden, B.W., Scarano, F., Watt, D.W. (2004). Assessment and application of quantitative schlieren methods: calibrated color schlieren and background oriented schlieren, *Exp. Fluids* **36** 309–25.
- [16] Goldhahn, E., Seume, J. (2007). The background oriented schlieren technique: sensitivity, accuracy, resolution and application to a three-dimensional density field, *Exp. Fluid* **43** 241–9.
- [17] Atcheson, B., Heidrich, W., Ihrke, I. (2008). An evaluation of optical flow algorithms for background oriented schlieren imaging, *Exp. Fluids* **46** 467–76.
- [18] Hargather, M.J., Settles, G.S. (2010). Natural-background oriented schlieren imaging, *Exp. Fluids* **48** 59–68.
- [19] Venkatakrisnan, L., Wiley, A., Kumar, R., Alvi, F. (2011). Density field measurements of a supersonic impinging jet with microjet control, *AIAA Journal* **49** 432–8.
- [20] Settles, G.S. (2018). Smartphone schlieren and shadowgraph imaging, *Optics and Lasers in Engineering* **104**: 9-21.
- [21] Hill, M. and Haering, E. (2016). Ground-to-air flow visualization using Solar Calcium-K line Background-Oriented Schlieren, *Experiments in Fluids*, 58(1).
- [22] Settles, G.S., Hargather M.J. (2017). A review of recent developments in schlieren and shadowgraph techniques, *Measurement Science and Technology* **28**(4): 042001.
- [23] Raffel, M. (2015). Background-oriented schlieren (BOS) techniques, *Exp. Fluids* **56** 60.

- [24] Friel, J., Grande, J. and Hetzner, D. (2000). Practical guide to image analysis, Materials Park (Ohio): ASM International.
- [25] Russ, J. (2011). The image processing handbook, Bosa Roca: CRC Press.
- [26] Woynar, L. (1998). Image Analysis: Applications in Materials Engineering, Boca Raton, Fla.: CRC Press.
- [27] Ce, L., W. T. Freeman, R. Szeliski and K. Sing Bing (2006). Noise Estimation from a Single Image, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06).
- [28] Thielicke, W., Stamhuis, E.J. (2014). PIVlab – Towards User – friendly, Affordable and Accurate Digital Particle Image Velocimetry in MATLAB, Journal of open research software, 2(E30).
DOI: <http://doi.org/10.5334/jors.bl>
- [29] Raffel, M., Willert, C., Wereley, S. and Kompenhans, J. (2007). Particle Image Velocimetry, Dordrecht: Springer.
- [30] Shavit, U., Lowe, R. and Steinbuck, J. (2006). Intensity Capping: a simple method to improve cross-correlation PIV results, Experiments in Fluids, 42(2), pp.225-240.
- [31] Pizer, S. M., E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman and K. Zuiderveld (1987). Adaptive histogram equalization and its variations, Computer Vision, Graphics, and Image Processing **39**(3): 355-368.
- [32] Otsuka, T., Wolanski, P. (2001). Particle image velocimetry (PIV) analysis of flame structure, Journal of Loss Prevention in the Process Industries, 14(1), pp. 503-507.
- [33] Yin, R., L. Xu and X. Li (2015). A simplified PIV-based method for flame velocity distribution measurement, 2015 IEEE International Conference on Imaging Systems and Techniques (IST).
- [34] F. Moisy, PIVMat, A PIV post-processing and data analysis toolbox for MATLAB, <http://www.fast.u-psud.fr/pivmat/> [Accessed 15 Jan. 2019].
- [35] <https://www.mathworks.com/matlabcentral/fileexchange/27659-pivlab-particle-image-velocimetry-piv-tool> [Accessed 10 Jan. 2019].

[36] Schneider, C.A., Rasband, W.S., Eliceiri, K.W. NIH image to ImageJ: 25 years of image analysis. Nat Meth 2012; 9:671-5 See also <https://imagej.net/ImageJ> [Accessed 2 Jan. 2019].

[37] YouTube. (2019). Rolling Shutter Explained (Why Do Cameras Do This?) - Smarter Every Day 172. [Online] Available at: <https://www.youtube.com/watch?v=dNVtMmLlnoE> [Accessed 30 May 2019].

[38] En.wikibooks.org. (2019). A-level Physics (Advancing Physics)/what is a wave? /Worked Solutions - Wikibooks, open books for an open world. [Online] Available at: [https://en.wikibooks.org/wiki/A-level_Physics_\(Advancing_Physics\)/What_is_a_wave%3F/Worked_Solutions](https://en.wikibooks.org/wiki/A-level_Physics_(Advancing_Physics)/What_is_a_wave%3F/Worked_Solutions) [Accessed 24 April 2019].

[39] Yang, W. (2001). Handbook of flow visualization. New York: Taylor and Francis.

[40] FutureLearn, (2019). Image analysis methods for biologists. [online] Available at: <https://www.futurelearn.com/courses/image-analysis> [Accessed 3 Jan. 2019].

[41] En.wikipedia.org. (2019). *Divergence*. [online] Available at: <https://en.wikipedia.org/wiki/Divergence> [Accessed 13 Mar. 2019].

[42] Betterexplained.com. (2019). *Vector Calculus: Understanding Divergence – Better Explained*. [online] Available at: <https://betterexplained.com/articles/divergence/> [Accessed 13 Mar. 2019].

Appendix A

MATLAB script for BOS dot pattern [34]:

```
“function makebospattern(n,diam,figtype,filename)”  
  
“%MAKEBOSPATTERN Random dot pattern for SS / BOS / FS-SS applications”  
  
“% MAKEBOSPATTERN(N,D) makes a figure filled with N randomly distributed”  
  
“% dots of diameter D for SS (Synthetic Schlieren), BOS (Background-”  
  
“% oriented Schlieren) or FS-SS (Free-Surface SS) applications.”  
  
“%”  
  
“% The figure format is portrait (ie, vertical) A4, 210x297 mm, and the”  
  
“% particle diameter D is in mm. Typical values are N=50000, D=1.”  
  
“%”  
  
“% MAKEBOSPATTERN(N,D,'b') makes black points on a white ground (by”  
  
“% default). MAKEBOSPATTERN(N,D,'w') makes white points on a black ground.”  
  
“%”  
  
“% MAKEBOSPATTERN(N,D,'.',FILENAME) saves the result in a 300-DPI TIFF”  
  
“% figure. Use the standard Windows viewer to print the figure.”  
  
“%”  
  
“% Notes:”  
  
“% - If you zoom the figure, the particles are not resized: particles”
```

“% are drawn in absolute units (points), while the paper size is in”
“% physical units (centimeters).”
“% - a 'particle' is a set of 4 concentric circles with increasing (or”
“% decreasing) gray levels. The 'particle diameter' D is the diameter of”
“% the outer circle; the inner circle has diameter 0.3*D. Depending on”
“% the printer quality, the outer circle may not print correctly, so the”
“% actual size may appear slightly larger or smaller than the requested”
“% size.”
“% - for FS-SS applications, see the function SURFHEIGHT for the”
“% surface height reconstruction.”
“% ”
“% Example:”
“% makebospattern(50000,1,'w','myfig');”
“% ”
“% See also PRINT, SURFHEIGHT”

“% F. Moisy, moisy_at_fast.u-psud.fr”
“% Revision: 1.21, Date: 2009/04/14.”

“% This function is part of the PIVMat Toolbox”

“% History:”

“% 2006/06/26: v1.00, first version ('makepidpattern')”

“% 2006/07/03: v1.10, S in mm; black/white option; saves in a TIFF file.”

“% 2006/07/05: v1.11, no need to set the PaperPosition and Background mode.”

“% 2007/05/29: v1.12, minor additional statistics”

“% 2008/09/05: v1.20, now entitled 'makebospattern'; included in PIVMat.”

“% 2009/04/14: v1.21, help text changed (BOS->SS)”

“% default values:”

“if ~exist('n','var'), n=50000; end”

“if ~exist('diam','var'), diam=0.25; end”

“if ~exist('figtype','var'), figtype='b'; end”

“h=gcf;”

```

“% Window size:”

“set(h,'Position',[360 80 560/sqrt(2) 560]);”

“set(gca,'Position',[0 0 1 1]); % location of the figure in the window”

“set(gca,'PlotBoxAspectRatio',[1/sqrt(2) 1 1]);”

“% Printing settings:”

“set(h,'PaperUnits','centimeters');”

“set(h,'PaperOrientation','portrait');”

“set(h,'PaperType','A4');”

“% set(h,'PaperSize',[21 29.7]);”

“set(h,'PaperPosition',[0 0 21 29.7]);”

“set(h,'PaperPositionMode','manual');”

“set(h,'InvertHardcopy','off'); % keep the user background mode”

“switch lower(figtype),”

“ case 'b' % black points on a white ground”

“ col{1} = 0.8*ones(1,3);”

“ col{2} = 0.6*ones(1,3);”

“ col{3} = 0.4*ones(1,3);”

```

```

“ col{4} = 0 * ones(1,3);”

“ set(h,'Color',[1 1 1]);”

“ case 'w' % white points on a black ground”

“ col{1} = 0.4*ones(1,3);”

“ col{2} = 0.6*ones(1,3);”

“ col{3} = 0.8*ones(1,3);”

“ col{4} = 1 * ones(1,3);”

“ set(h,'Color',[0 0 0]);”

“end”

“x=rand(1,n);”

“y=rand(1,n);”

“s = diam * 72/25.4; % diameter, in points units (1 point = 1/72 inch = 25.4/72 mm)”

“plot(x,y,'o','MarkerFaceColor',col{1},'MarkerEdgeColor',col{1},'Markersize',s);”

“hold on”

“plot(x,y,'o','MarkerFaceColor',col{2},'MarkerEdgeColor',col{2},'Markersize',0.7*s);”

“plot(x,y,'o','MarkerFaceColor',col{3},'MarkerEdgeColor',col{3},'Markersize',0.5*s);”

“plot(x,y,'o','MarkerFaceColor',col{4},'MarkerEdgeColor',col{4},'Markersize',0.3*s);”

```

```
“hold off”
```

```
“axis off;”
```

```
“%display some info:”
```

```
“disp(' ');”
```

```
“disp('PID pattern parameters:');”
```

```
“switch lower(figtype)”
```

```
“ case 'b’”
```

```
“ disp([' ' num2str(n) ' black particles on a 210x297 mm (A4) paper']);”
```

```
“ case 'w’”
```

```
“ disp([' ' num2str(n) ' white particles on a 210x297 mm (A4) paper']);”
```

```
“end”
```

```
“disp([' Particle diameter = ' num2str(diam) ' mm.']);”
```

```
“disp([' ' num2str(n/(210*297)) ' particles / mm^2']);”
```

```
“disp([' filled surface ratio = ' num2str((n*pi*diam^2/4)/(297*210))]);”
```

```
“disp(' ');”
```

```
“disp('Assuming a 1280x1024 camera, with 210 mm = 1024 pixels:');”
```

```
“pdiam = diam/210*1024;”
```

```

“disp([' particle diameter = ' num2str(pdiam) ' pixels']);”

“if pdiam<1.3”

“ disp(' * Warning: particles too small *');”

“elseif pdiam>4”

“ disp(' * Warning: particles too big *');”

“end”

“ncam = n * (5/4) / sqrt(2); % number of particles in a 5/4 camera field (eg, 1280x1024).”

“ppp = ncam / (1280*1024);”

“disp([' ' num2str(ppp) ' particles / pixel^2']);”

“if ppp>0.2”

“ disp(' * Warning: high density *');”

“elseif ppp<0.02”

“ disp(' * Warning: low density *');”

“end”

“disp(' ');”

“disp('Particles per interrogation window:');”

“winsize = [6 8 12 16 32 64];”

“for nws=1:length(winsize)”

```

```

“ ppw (nws) = winsize(nws)^2 * ncam / (1280*1024);”

“ disp([' ' num2str(ppw (nws)) ' particles / ' num2str(winsize(nws)) '^2-window']);”

“end”

“optws = sqrt(5*(1280*1024)/ncam);”

“disp([' Optimal window size: ' num2str(optws,3) '^2']);”

“dif=abs(ppw-5);”

“ind=find(dif==min(dif));”

“disp([' Closest window size: ' num2str(winsize(ind)) '^2']);”

“disp(' ');”

“% save the figure:”

“if exist('filename','var’)”

“ print('-dtiff','-r500',filename);”

“ disp(['"" filename "" saved']);”

“end”

```

Appendix B

MATLAB script for seeding particles with dot pattern [34]:

```
“%function makebospattern(n,diam,figtype,filename)”

“%MAKEBOSPATTERN Random dot pattern for SS / BOS / FS-SS applications”

“% MAKEBOSPATTERN(N,D) makes a figure filled with N randomly distributed”

“% dots of diameter D for SS (Synthetic Schlieren), BOS (Background-”

“% oriented Schlieren) or FS-SS (Free-Surface SS) applications.”

“%”

“% The figure format is portrait (ie, vertical) A4, 210x297 mm, and the”

“% particle diameter D is in mm. Typical values are N=50000, D=1.”

“%”

“% MAKEBOSPATTERN(N,D,'b') makes black points on a white ground (by”

“% default). MAKEBOSPATTERN(N,D,'w') makes white points on a black ground.”

“%”

“% MAKEBOSPATTERN(N,D,'!',FILENAME) saves the result in a 300-DPI TIFF”

“% figure. Use the standard Windows viewer to print the figure.”

“%”

“% Notes:”
```

```
“% - If you zoom the figure, the particles are not resized: particles”  
“% are drawn in absolute units (points), while the paper size is in”  
“% physical units (centimeters).”  
“% - a 'particle' is a set of 4 concentric circles with increasing (or”  
“% decreasing) gray levels. The 'particle diameter' D is the diameter of”  
“% the outer circle; the inner circle has diameter 0.3*D. Depending on”  
“% the printer quality, the outer circle may not print correctly, so the”  
“% actual size may appear slightly larger or smaller than the requested”  
“% size.”  
“% - for FS-SS applications, see the function SURFHEIGHT for the”  
“% surface height reconstruction.”  
“%”  
“% Example:”  
“% makebospattern(50000,1,'w','myfig');”  
“%”  
“% See also PRINT, SURFHEIGHT”  
  
“% F. Moisy, moisy_at_fast.u-psud.fr”
```



```
“% Revision: 1.21, Date: 2009/04/14.”
```

```
“% This function is part of the PIVMat Toolbox”
```

```
“% History:”
```

```
“% 2006/06/26: v1.00, first version ('makepidpattern')”
```

```
“% 2006/07/03: v1.10, S in mm; black/white option; saves in a TIFF file.”
```

```
“% 2006/07/05: v1.11, no need to set the PaperPosition and Background mode.”
```

```
“% 2007/05/29: v1.12, minor additional statistics”
```

```
“% 2008/09/05: v1.20, now entitled 'makebospattern'; included in PIVMat.”
```

```
“% 2009/04/14: v1.21, help text changed (BOS->SS)”
```

```
“clc”
```

```
“clear all”
```

```
“% default values:”
```

```
“if ~exist('n','var'), n=5000; end”
```

```
“%if ~exist('diam','var'), diam=0.25; end”
```

```
“if ~exist('diam','var'), diam=1; end”
```

```
“if ~exist('figtype','var'), figtype='b'; end”

“n”

“h=gcf;”

“% Window size:”

“set(h,'Position',[360 80 560/sqrt(2) 560]);”

“set(gca,'Position',[0 0 1 1]); % location of the figure in the window”

“set(gca,'PlotBoxAspectRatio',[1/sqrt(2) 1 1]);”

“% Printing settings:”

“set(h,'PaperUnits','centimeters');”

“set(h,'PaperOrientation','portrait');”

“set(h,'PaperType','A4');”

“% set(h,'PaperSize',[21 29.7]);”

“set(h,'PaperPosition',[0 0 21 29.7]);”

“set(h,'PaperPositionMode','manual');”

“set(h,'InvertHardcopy','off'); % keep the user background mode”

“switch lower(figtype),”
```

```
“ case 'b' % black points on a white ground”
```

```
“ col{1} = 0.8*ones(1,3);”
```

```
“ col{2} = 0.6*ones(1,3);”
```

```
“ col{3} = 0.4*ones(1,3);”
```

```
“ col{4} = 0 * ones(1,3);”
```

```
“ set(h,'Color',[1 1 1]);”
```

```
“ case 'w' % white points on a black ground”
```

```
“ col{1} = 0.4*ones(1,3);”
```

```
“ col{2} = 0.6*ones(1,3);”
```

```
“ col{3} = 0.8*ones(1,3);”
```

```
“ col{4} = 1 * ones(1,3);”
```

```
“ set(h,'Color',[0 0 0]);”
```

```
“end”
```

```
“x=rand(1,n);”
```

```
“y=rand(1,n);”
```

```
“%Seeding particles:”
```

```
“xsA = rand(1,1000);”
```

```
“ysA = rand(1,1000)*0.9;”
```

```
“%Shift seeding particles A along the y-axis:”
```

```
“xsB = xsA;”
```

```
“ysB = ysA + 0.01;”
```

```
“s = diam * 72/25.4; % diameter, in points units (1 point = 1/72 inch = 25.4/72 mm)”
```

```
“figure(1)”
```

```
“plot(x,y,'o','MarkerFaceColor','k','MarkerEdgeColor','k','Markersize',s);”
```

```
“% plot(x,y,'o','MarkerFaceColor',col{1},'MarkerEdgeColor',col{1},'Markersize',s);”
```

```
“% hold on”
```

```
“% plot(x,y,'o','MarkerFaceColor',col{2},'MarkerEdgeColor',col{2},'Markersize',0.7*s);”
```

```
“% plot(x,y,'o','MarkerFaceColor',col{3},'MarkerEdgeColor',col{3},'Markersize',0.5*s);”
```

```
“% plot(x,y,'o','MarkerFaceColor',col{4},'MarkerEdgeColor',col{4},'Markersize',0.3*s);”
```

```
“% hold off”
```

```
“axis off;”
```

```
“% *****”
```

```
“% Window size:”
```

```

“figure(2)”

“h2=gcf;”

“set(h2,'Position',[360 80 560/sqrt(2) 560]);”

“set(gca,'Position',[0 0 1 1]); % location of the figure in the window”

“set(gca,'PlotBoxAspectRatio',[1/sqrt(2) 1 1]);”

“hold on”

“plot(x,y,'o','MarkerFaceColor','k','MarkerEdgeColor','k','Markersize',s);”

“plot(xsA,ysA,'o','MarkerFaceColor','r','MarkerEdgeColor','r','Markersize',s);”

“hold off”

“axis off;”

“% *****”

“figure(3)”

“h3=gcf;”

“set(h3,'Position',[360 80 560/sqrt(2) 560]);”

“set(gca,'Position',[0 0 1 1]); % location of the figure in the window”

“set(gca,'PlotBoxAspectRatio',[1/sqrt(2) 1 1]);”

“hold on”

“plot(x,y,'o','MarkerFaceColor','k','MarkerEdgeColor','k','Markersize',s);”

“plot(xsB,ysB,'o','MarkerFaceColor','r','MarkerEdgeColor','r','Markersize',s);”

```

```

“hold off”

“axis off;”

“%display some info;”

“disp(' ');”

“disp('PID pattern parameters:');”

“switch lower(figtype)”

“ case 'b’”

“     disp([' ' num2str(n) ' black particles on a 210x297 mm (A4) paper']);”

“ case 'w’”

“     disp([' ' num2str(n) ' white particles on a 210x297 mm (A4) paper']);”

“end”

“disp([' Particle diameter = ' num2str(diam) ' mm.']);”

“disp([' ' num2str(n/(210*297)) ' particles / mm^2]);”

“disp([' filled surface ratio = ' num2str((n*pi*diam^2/4)/(297*210))]);”

“disp(' ');”

“disp('Assuming a 1280x1024 camera, with 210 mm = 1024 pixels:');”

“pdiam = diam/210*1024;”

“disp([' particle diameter = ' num2str(pdiam) ' pixels']);”

```

```

“if pdiam<1.3”

“ disp(' * Warning: particles too small *');”

“elseif pdiam>4”

“ disp(' * Warning: particles too big *');”

“end”

“ncam = n * (5/4) / sqrt(2); % number of particles in a 5/4 camera field (eg, 1280x1024).”

“ppp = ncam / (1280*1024);”

“disp([' ' num2str(ppp) ' particles / pixel^2']);”

“if ppp>0.2”

“ disp(' * Warning: high density *');”

“elseif ppp<0.02”

“ disp(' * Warning: low density *');”

“end”

“disp(' ');”

“disp('Particles per interrogation window:');”

“winsize = [6 8 12 16 32 64];”

“for nws=1:length(winsize)”

“ ppw (nws) = winsize(nws)^2 * ncam / (1280*1024);”

```

```
“ disp([' ' num2str(ppw (nws)) ' particles / ' num2str(winsize(nws)) '^2-window']);”
```

```
“end”
```

```
“optws = sqrt(5*(1280*1024)/ncam);”
```

```
“disp([' Optimal window size: ' num2str(optws,3) '^2']);”
```

```
“dif=abs(ppw-5);”
```

```
“ind=find(dif==min(dif));”
```

```
“disp([' Closest window size: ' num2str(winsize(ind)) '^2']);”
```

```
“disp(' ');”
```

```
“% save the figure:”
```

```
“if exist('filename','var’)”
```

```
“ print('-dtiff','-r500',filename);”
```

```
“ disp(['"" filename "" saved']);”
```

```
“end”
```


Appendix C

MATLAB script for absolute difference images:

%analyse images by taking difference of any image by the reference

```
clear; close all; clc
```

```
I = imread('r.bmp');
```

%images upto 52 is at start of flame and then after 10 sec upto 106

```
for i=1:41
```

```
    filn = [int2str(i) '.bmp']
```

```
    %J = imread('40.bmp');
```

```
    J = imread(filn);
```

```
    K = imabsdiff(I,J)*8;
```

```
    figure(1), imshow(K)
```

```
    pause(0.2)
```

```
end
```

```
K2=K(129:700,75:400)
```

```
figure(2)
```

`imshow(K2)`

Appendix D

MATLAB script for video split:

```
“%%Extracting & Saving of frames from a Video file through Matlab Code%%”
```

```
“clc;”
```

```
“close all;”
```

```
“clear all;”
```

```
“% assigning the name of sample avi file to a variable”
```

```
“filename = 'slow video 1.mp4';”
```

```
“%reading a video file”
```

```
“mov = VideoReader(filename);”
```

```
“% Defining Output folder as 'snaps”
```

```
“opFolder = fullfile(cd, 'snaps');”
```

```
“%if not existing”
```

```
“if ~exist(opFolder, 'dir')”
```

```
“%make directory & execute as indicated in opfolder variable”
```

```
“mkdir(opFolder);”
```

“end”

“%getting no of frames”

“numFrames = mov.NumberOfFrames;”

“%setting current status of number of frames written to zero”

“numFramesWritten = 0;”

“%for loop to traverse & process from frame '1' to 'last' frames”

“for t = 1 : numFrames”

“currFrame = read(mov, t); %reading individual frames”

“opBaseFileName = sprintf('%3.3d.bmp', t);”

“opFullFileName = fullfile(opFolder, opBaseFileName);”

“imwrite(currFrame, opFullFileName, 'bmp'); %saving as 'png' file”

“%indicating the current progress of the file/frame written”

“progIndication = sprintf('Wrote frame %4d of %d.', t, numFrames);”

“disp(progIndication);”

“numFramesWritten = numFramesWritten + 1;”

“end %end of 'for' loop”

```
“progIndication = sprintf('Wrote %d frames to folder "%s"',numFramesWritten, opFolder);”
```

```
“disp(progIndication);”
```

```
“%End of the code”
```