

## **Real-Time Optical Flow Velocimetry for Natural and Built Environments**

**Luís Palma Nunes Mendes**

Supervisor: Doctor Alexandre José Malheiro Bernardino  
Co-Supervisors: Doctor Rui Miguel Lage Ferreira,  
Doctor Ana Margarida da Costa Ricardo

Thesis approved in public session to obtain the PhD Degree in

**Electrical and Computer Engineering**

Jury final classification

**Pass with Distinction**

**2024**



**UNIVERSIDADE DE LISBOA**  
**INSTITUTO SUPERIOR TÉCNICO**

**Real-Time Optical Flow Velocimetry for Natural and Built  
Environments**

**Luís Palma Nunes Mendes**

**Supervisor:** Doctor Alexandre José Malheiro Bernardino

**Co-Supervisors:** Doctor Rui Miguel Lage Ferreira  
Doctor Ana Margarida da Costa Ricardo

Thesis approved in public session to obtain the PhD Degree in

**Electrical and Computer Engineering**

Jury final classification

**Pass with Distinction**

Jury

**Chairperson:** **Doctor Paulo Jorge Coelho Ramalho Oliveira,**  
Instituto Superior Técnico, Universidade de Lisboa

**Members of the Committee:**

**Doctor Tianshu Liu,**  
Western Michigan University, USA

**Doctor José Alberto Rosado dos Santos Victor,**  
Instituto Superior Técnico, Universidade de Lisboa

**Doctor Rui Miguel Lage Ferreira,**  
Instituto Superior Técnico, Universidade de Lisboa

**Doctor João Miguel Fernandes Rodrigues,**  
Instituto Superior de Engenharia, Universidade do Algarve

**Doctor Rui Jorge Ferreira Aleixo,**  
Civil Engineering Research and Innovation for Sustainability (CERIS)

**Funding Institutions:**

**Fundação para a Ciência e Tecnologia (FCT) - SFRH/BD/137967/2018**

**2024**





Dedicated to my parents Ana e Casimiro for their encouragement and patience throughout all my life.



### Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.



## Acknowledgments

Special acknowledgment goes to my parents Ana e Casimiro, given that it was through their patience and support that this PhD was made possible and got this far.

I thank my supervisors Prof. Alexandre Bernardino, Prof. Rui Ferreira and Dr. Ana Margarida Ricardo, for their advices, project coordination, as well as, for their revision work over the multiple conference papers and journal articles, and also for their friendship. Um forte Abraço.

My family, friends, colleagues and professors for their support and experience exchange. Special thanks go to my friend Alberto Machado, in appreciation for the help he gave me with his lathe machine, the stencil printer, soldering iron tooling and acrylic cutting.

I thank Fundação da Ciência e Tecnologia (Portuguese Foundation for Science and Technology) for supporting my PhD through the PhD grant SFRH/BD/137967/2018.



## Resumo

Esta tese investiga a aplicação de métodos de fluxo óptico (OpF) em Velocimetria por Imagem de Partículas (PIV), fazendo a análise e comparação de combinações inovadoras de OpF. Destaca-se a sinergia resultante da combinação dos métodos Liu-Shen e Lucas-Kanade, bem como do Liu-Shen com o Farnebäck. Os métodos OpF de Lucas-Kanade e Farnebäck, comuns na visão computacional, são revisitados para mecânica dos fluídos, onde Farnebäck tem sido negligenciado. Avalia-se um novo método de PIV híbrido que integra Liu-Shen combinado com Lucas-Kanade, focando-se o estudo na recuperação de informação de turbulência e melhoria do desempenho do PIV. A pesquisa inclui o lançamento de softwares de código aberto, como um gerador de imagens de PIV sintéticas, implementações de referência de métodos de OpF e o QuickLabPIV-ng, um software Java acelerado por Aparapi/OpenCL e com Interface Gráfica de Utilizador (GUI). Testa-se também uma unidade de iluminação LED PIV portátil, de custo acessível, criada de raiz e que procura incentivar o contacto antecipado com o PIV nas licenciaturas, juntamente com o software QuickLabPIV-ng. Os métodos OpF e PIV híbridos são rigorosamente testados contra bases de dados de imagens PIV sintéticas e reais, com diferentes qualidades e características. Avalia-se o desempenho dos métodos em vários tipos de escoamentos, considerando as características das imagens de PIV, como o ruído, tamanho da partícula, ou a concentração, entre outros. Os resultados indicam que o método híbrido PIV-OpF melhora a resolução espacial relativamente ao PIV, permitindo a análise de turbulência de escalas menores, embora necessitando de imagens PIV de boa qualidade. O estudo sugere que o OpF pode substituir as técnicas tradicionais de PIV e mapeia regiões de escoamento críticas e condições de imagem que afetam a precisão dos métodos.

**Palavras-chave:** PIV, Fluxo Óptico, PIV Híbrido, GpGPU, GUI





## Abstract

This thesis delves into the integration of optical flow (OpF) methods within Particle Image Velocimetry (PIV) applications, presenting an innovative analysis and benchmarking of novel OpF combinations. Specifically, it explores the synergy of Liu-Shen with Lucas-Kanade and Liu-Shen with Farnebäck. The study revisits the Lucas-Kanade and Farnebäck OpF methods, traditionally rooted in computer vision, and introduces the latter to fluid mechanics applications, a domain where it has been previously overlooked. A novel hybrid PIV approach, combining Liu-Shen with Lucas-Kanade, is assessed for its efficacy in turbulence information retrieval and enhancement of overall PIV performance. The research is complemented by the release of several open-source software tools to the public domain, including a synthetic PIV image generator, a comprehensive suite of OpF method reference implementations, and QuickLabPIV-ng—an efficient Aparapi/OpenCL implementation equipped with a Graphical User Interface (GUI) in Java. Additionally, an economical, portable PIV LED lighting unit is tested, that was created from scratch, promoting the early adoption of PIV technology in academic settings in tandem with QuickLabPIV-ng software. The OpF and hybrid PIV methodologies undergo rigorous benchmarking against both synthetic PIV image databases with established ground truth and real PIV image databases exhibiting varied quality and characteristics. The performance of these methods is meticulously evaluated across different flow types, considering the specific characteristics of PIV images to optimize method efficacy. The findings suggest that the hybrid PIV-OpF method enhances spatial resolution and augments the capacity of PIV to yield data suitable for fine-scale turbulence analysis, contingent upon the high quality of PIV images. Furthermore, the study posits OpF as a viable alternative to conventional PIV techniques. It also delineates critical flow regions and image conditions that influence method's accuracies.

**Keywords:** PIV, Optical Flow, Hybrid PIV, GpGPU, GUI



# Contents

|   |          |
|---|----------|
| Acknowledgments . . . . .   | ix       |
| Resumo . . . . .  | xi       |
| Abstract . . . . .  | xiii     |
| List of Tables . . . . .  | xix      |
| List of Figures . . . . .   | xxiii    |
| Nomenclature . . . . .  | xxxi     |
| Glossary . . . . .  | xxxiii   |
| <b>Chapter 1: Introduction</b>  | <b>1</b> |
| 1.1 Motivation and research questions . . . . .   | 1        |
| 1.2 Objectives and Deliverables . . . . .   | 2        |
| 1.3 Impact . . . . .  | 3        |
| 1.3.1 Potential for technology transfer . . . . .                                       | 3        |
| 1.3.2 Benefits to research institutions . . . . .                                       | 4        |
| 1.4 Thesis Outline . . . . .  | 4        |
| 1.5 Related works . . . . .   | 5        |
| <b>Chapter 2: An image generating tool for planar PIV and Optical Flow benchmarking</b> | <b>7</b> |
| 2.1 Abstract . . . . .  | 7        |
| 2.2 Motivation and significance . . . . .   | 8        |
| 2.3 Software description . . . . .  | 9        |
| 2.3.1 General description . . . . .   | 9        |
| 2.3.2 Flow types . . . . .  | 10       |
| 2.3.3 Software Architecture . . . . .   | 11       |
| 2.3.4 Image parameterization and generation . . . . .                                   | 13       |
| 2.4 Illustrative Examples . . . . .   | 15       |
| 2.5 Impact . . . . .  | 19       |
| 2.6 Conclusions . . . . .   | 20       |
| 2.7 Credit . . . . .  | 21       |
| 2.8 Conflict of Interest . . . . .  | 21       |
| 2.9 References . . . . .  | 21       |

|  |           |
|--|-----------|
| <b>Chapter 3: A comparative study of optical flow methods for fluid mechanics</b>      | <b>25</b> |
| 3.1 Abstract . . . . .   | 25        |
| 3.2 Introduction . . . . .   | 26        |
| 3.3 Synthetic image and flow generator . . . . .                                       | 30        |
| 3.3.1 Flow field generation . . . . .  | 30        |
| 3.3.2 Synthetic image generation . . . . .   | 30        |
| 3.4 Benchmark – data, algorithms and metrics . . . . .                                 | 34        |
| 3.4.1 Benchmark data . . . . .   | 34        |
| 3.4.2 Methods and their implementation . . . . .                                       | 34        |
| 3.4.3 Error metrics . . . . .  | 34        |
| 3.4.4 Quantification of intrinsic parameters of OpF methods and PIV image analysis . . | 36        |
| 3.5 Results . . . . .  | 37        |
| 3.5.1 Pyramidal vs non-pyramidal optical flow and bit-depths . . . . .                 | 37        |
| 3.5.2 Combination with Liu-Shen (2008) and effect of image bit-depth . . . . .         | 37        |
| 3.5.3 Performance of OpF methods according to flow type . . . . .                      | 41        |
| 3.5.4 Influence of imaged particle spot sizes and particle number density . . . . .    | 44        |
| 3.6 Velocity field analysis . . . . .  | 45        |
| 3.7 Discussion . . . . .   | 49        |
| 3.8 Conclusion . . . . .   | 51        |
| 3.9 Acknowledgements . . . . .   | 53        |
| Appendices   |           |
| 3.A Appendices . . . . .   | 53        |
| 3.A.1 PIV and optical flow methods . . . . .   | 53        |
| 3.A.2 Pyramidal vs non-pyramidal optical flow and bit-depths . . . . .                 | 55        |
| 3.11 References . . . . .  | 60        |
| <b>Chapter 4: QuickLabPIV new generation software</b>                                  | <b>67</b> |
| 4.1 Introduction . . . . .   | 67        |
| 4.2 Software Workflow and Main Features . . . . .                                      | 67        |
| 4.2.1 Workflow and Hybridization Options . . . . .                                     | 67        |
| 4.2.2 PIV Key Steps—Cross-Correlation and Sub-Pixel Interpolation . . . . .            | 69        |
| 4.2.3 OpF Methods . . . . .  | 69        |
| 4.2.4 Hybridization of OpF and PIV after Correlation Steps . . . . .                   | 69        |
| 4.2.5 Vector Inheritance and Warping . . . . .   | 70        |
| 4.2.6 Vector Validation and Substitution . . . . .                                     | 70        |
| 4.3 Software and Graphical User Interface design . . . . .                             | 71        |
| 4.3.1 Software design . . . . .  | 71        |
| 4.3.2 Graphical User Interface . . . . .   | 74        |
| 4.4 Computational Performance . . . . .  | 75        |

|                     |  |            |
|---------------------|--|------------|
| 4.5                 | References . . . . .   | 80         |
| <b>Chapter 5:</b>   | <b>A Hybrid PIV/Optical Flow Method for Incompressible Turbulent Flows</b> | <b>81</b>  |
| 5.1                 | Introduction . . . . .   | 81         |
| 5.2                 | Verification and Validation of the Hybrid PIV-OpF Software . . . . .       | 84         |
| 5.2.1               | Strategy . . . . .   | 84         |
| 5.2.2               | Synthetic Data . . . . .   | 85         |
| 5.2.3               | Laboratory Databases . . . . .   | 94         |
| 5.3                 | Methods and Materials . . . . .  | 102        |
| 5.4                 | Conclusions and Recommendations for Further Analysis . . . . .             | 103        |
| 5.5                 | References . . . . .   | 104        |
| <b>Chapter 6:</b>   | <b>Conclusions</b>   | <b>109</b> |
| 6.1                 | Achievements . . . . .   | 110        |
| 6.2                 | Future Work . . . . .  | 111        |
| <b>Appendix A:</b>  | <b>Portable PIV LED lighting unit and experimental test</b>                | <b>113</b> |
| A.1                 | Characteristics . . . . .  | 113        |
| A.2                 | Experimental PIV data from LED unit . . . . .                              | 116        |
| A.3                 | Conclusions . . . . .  | 116        |
| <b>Bibliography</b> |  | <b>121</b> |



# List of Tables

|     |  |    |
|-----|--|----|
| 2.1 | Code metadata (mandatory) . . . . .  | 8  |
| 2.2 | Flow types, velocity flow fields and displacement equations. . . . .   | 12 |
| 2.3 | Configurable control parameters . . . . .  | 15 |
| 3.1 | Flow types, velocity flow fields and displacement equations. The displacements in the flow field are analytically solved from the corresponding velocity flow field expression. An exception is made for the Rankine vortex with superimposed uniform flow, for which the solution is obtained numerically by a 4 <sup>th</sup> order Runge-Kutta scheme. In these equations, $x$ and $y$ are Cartesian orthogonal spatial coordinates specifying the position of a particle in a plane, $x_0$ and $y_0$ are the initial particle positions, $r$ is the radial polar coordinate, $\theta$ is the azimuthal polar coordinate, $\theta_0$ is an initial azimuthal particle position, $t$ is the time elapsed since an initial instant ( $t = 0$ ), $x_c$ and $y_c$ are the coordinates of the stagnation point, for the inviscid stagnation point flow, or the centre of the vortex, for Rankine vortices, $\Gamma$ is the circulation of rotational core in the Rankine vortices, $y_{max}$ and $y_{min}$ are the coordinates of the upper and lower bounds of the flux tube within which the Poiseuille flow takes place, $R$ is the radius for which maximum azimuthal velocity is attained in the Rankine vortex flows, $m$ is a reference velocity, $M$ is a scaling parameter that allows for the specification of the desired maximum relative displacement (a percentage of the IA size, always less than or equal to 25%). For the Rankine Vortex and Rankine with superimposed Uniform flow cases $\Gamma = m2\pi R$ . . . . . | 31 |
| 3.2 | Configurable image control parameters. Here, $d_\tau$ is the spot diameter of the imaged tracer particle, $N_i$ is the number of initial in-plane particles in each interrogation area (particles whose centres are within the virtual laser sheet thickness), $\Delta D$ is the maximum in-plane displacement, $b$ is the image bit-depth, $N$ is the power of the White Gaussian Image noise (WGIN), $\sigma_Z$ represents the standard deviation of the out-of-plane displacement (whose mean is zero), $A^2$ is the image resolution, $\Delta Z_0$ is the laser sheet thickness, $I_0$ is the particle reflection peak intensity, $f_u$ is the image scale factor, and $IA$ is the size of the interrogation area for PIV measurements. The laser sheet thickness, $\Delta Z_0$ , is defined as the normal distance between two parallel planes, on either side of the central plane, for which the light intensity is $1/e^2$ of the peak (central) laser light intensity. . . . .  | 32 |

|     |   |    |
|-----|---|----|
| 3.3 | Values of the Lagrange multiplier ( $\lambda$ ) for the Horn-Schunck method as a function of pyramidal level ( $l$ ), image bit-depth ( $b$ ) and number of particles per IA volume ( $N_i$ ). . . . .  | 36 |
| 3.4 | Absolute and relative errors (Avg.) and standard deviations (Std.) of the base algorithms and of the base algorithms combined with the Liu-Shen method. All statistics are discriminated by bit depth and are computed from SIG ensembles. Each SIG ensemble is composed of all flow types, particle spot sizes, particle concentrations and maximum displacements. All methods feature a pyramidal 2-level pass. . . . . | 38 |
| 3.5 | Absolute and relative errors (Avg.) and standard deviations (Std.) of the base algorithms and of the base algorithms combined with 2-level pyramidal processing. All statistics are discriminated by displacement and are computed from SIG ensembles. Each SIG ensemble is composed of all flow types, particle spot sizes, particle concentrations and image bit depths. . . . .  | 55 |
| 3.6 | Pyramidal L2 dense Lucas-Kanade improvements over dense Lucas-Kanade per flow type/displacement and bit depth. Where $E$ is the SIG ensemble absolute error and $E_R$ is the SIG ensemble relative error. . . . .   | 56 |
| 3.7 | Pyramidal L2 Farnebäck improvements over Farnebäck per flow type/displacement and bit depth. Where $E$ is the SIG ensemble absolute error and $E_R$ is the SIG ensemble relative error. . . . .   | 57 |
| 3.8 | Pyramidal L2 Horn-Schunck improvements over Horn-Schunck per flow type/displacement and bit depth. Where $E$ is the SIG ensemble absolute error and $E_R$ is the SIG ensemble relative error. . . . .   | 58 |
| 3.9 | Absolute and relative errors (Avg.) and standard deviations (Std.) of the base algorithms combined with Liu-Shen without pyramidal processing versus with 2-level pyramidal processing. All statistics are discriminated by displacement and are computed from SIG ensembles. Each SIG ensemble is composed of all flow types, particle spot sizes, particle concentrations and image bit depths. . . . .                 | 59 |
| 4.1 | QuickLabPIV-ng computation times for Platform1. . . . .   | 78 |
| 4.2 | QuickLabPIV-ng computation times for Platform2 with Linux. . . . .  | 78 |
| 4.3 | QuickLabPIV-ng computation times for Platform2 with Windows. . . . .  | 79 |
| 5.1 | Averaged absolute and relative errors for the Poiseuille flow for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume. . . . .   | 89 |
| 5.2 | Averaged absolute and relative errors for the Rankine vortex for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume. . . . .  | 91 |
| 5.3 | Averaged absolute and relative errors for the Rankine vortex with a superimposed uniform flow for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume. . . . .   | 92 |



|     |   |    |
|-----|---|----|
| 5.4 | Averaged absolute and relative errors for the inviscid flow with a stagnation point for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume. . . . . | 92 |
| 5.5 | Averaged absolute and relative errors for a uniform flow for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume.                                    | 92 |



# List of Figures

|     |   |    |
|-----|---|----|
| 2.1 | Flowchart diagram of PIV image generator tool . . . . .   | 11 |
| 2.2 | Flow types . . . . .  | 16 |
| 2.3 | Different particles sizes, same concentration . . . . .   | 17 |
| 2.4 | Same particles size, different concentrations . . . . .   | 17 |
| 2.5 | Particle concentration and Noise effect . . . . .   | 18 |
| 3.1 | Streamlines for the Poiseuille and inviscid stagnation point flows. Flow fields generated with the corresponding equations shown in Table 3.1. Thicker lines indicate higher velocities. Color code varies from dark blue (low velocity) to yellow (high velocities). All velocities are normalized by the highest velocity in each flow type. . . . .  | 32 |
| 3.2 | PIV image samples (164 px x 164 px) generated with the PIV Image Generator [61]. a) Imaged particle spot sizes of 6.0 px diameter and $N_i$ value of 1, b) imaged particle spot sizes of 3.0 px diameter and $N_i$ value of 16 . . . . .  | 33 |
| 3.3 | Distribution of absolute (left) and relative (right) errors for the base algorithms and for the base algorithms combined with the Liu-Shen method, 8-bit images. Top row: Horn-Schunck; middle row: Lucas-Kanade; bottom row: Farnebäck. All implementations comprise a pyramidal 2-level pass. . . . .   | 39 |
| 3.4 | Distribution of absolute (left) and relative (right) errors for the Farnebäck algorithm and for the same algorithm combined with the Liu-Shen method, 12-bit images. The implementation comprises a pyramidal 2-level pass. . . . .   | 40 |
| 3.5 | Direct comparison of OpF combinations for 8-bit images in terms of absolute errors (left column) and relative errors (right column). Top row shows the comparison between Horn-Shunck/Liu-Shen and Lucas-Kanade/Liu-Shen. Middle row shows the comparison between Horn-Shunck/Liu-Shen and Farnebäck/Liu-Shen. Bottom row shows the comparison between Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen. All methods include a pyramidal 2-level scheme. Boxplots condense the distribution of comparison errors, defined by equation (3.7). Each box encompasses the second and third quartiles. The whiskers encompass a band of $3IQR$ centered around the mean of the distribution. Circles represent values lying beyond this band. The median of the distribution is represented by the red bar in the box. . . . . | 42 |

|      |   |    |
|------|---|----|
| 3.6  | Direct comparison of the Farnebäck/Liu-Shen and Lucas-Kanade/Liu-Shen combinations for 8-bit images in terms of absolute errors (left plot) and relative errors (right plot). Both methods include a pyramidal 2-level scheme. Box-plots condense the distribution of comparison errors, defined by equation (3.7) Reminder of caption as in Figure 3.5. . . . .  | 44 |
| 3.7  | Accuracy vs particle spot size, WGIN and $N_i$ for Rankine vortex with superimposed uniform flow at 4.0 px max. displacement, for all 8, 10 and 12-bit image depths - Results shown for Lucas-Kanade/Liu-Shen combination, Farnebäck/Liu-Shen combination and PIV (Legend: $\times$ - denotes 1 px diameter imaged particle spot size, $+$ - denotes 2 px diameter imaged particle spot size, $\bigcirc$ - denotes 3 px diameter imaged particle spot size, $\star$ - denotes 6 px diameter imaged particle spot size, Black - denotes 0 dBW WGIN, Green - denotes 5 dBW WGIN, Red - denotes 15 dBW WGIN). . . . .  | 44 |
| 3.8  | Streamlines for the flow with a single stagnation point, located at $x = 250$ px, $y = 0$ px (see also Figure 3.1). Flow direction from top to bottom. The parameters for image generation were: maximum displacement equal to 4.0 px, WGIN set to 0 dbW, diameter of the imaged particle spot equal to 3.0 px, particle number concentration per IA equal to 6, out-of-plane standard deviation equal to 0.025 px. Results are shown for the theoretical (black thick line), correlation-based PIV (blue thin line), full resolution Lucas-Kanade/Liu-Shen combination (red dashed line) and subsampled Lucas-Kanade/Liu-Shen combination (red thin line), to match the PIV resolution. Boundary conditions to all streamlines are located at $y = 500$ px. . . . .  | 47 |
| 3.9  | Streamlines for the Poiseuille flow. Flow direction from the right to the left. Remaining caption as in Figure 3.8. . . . .   | 48 |
| 3.10 | Streamlines for the Rankine Vortex with superimposed uniform flow. The vortex flow is direct (anti-clockwise) while the uniform flow is from bottom left to upper-right at 45 deg. Remaining caption as in Figure 3.8. . . . .  | 48 |
| 3.11 | Distribution of relative errors expressing the accuracy of OpF methods <u>per</u> flow type in their optimal imaging conditions: $2.0 \leq d_\tau \leq 6.0$ [px], $6 \leq N_i \leq 12$ and 4 px maximum displacement. The assessment of accuracy is based on the empirical probability density of SIG errors. Each SIG error is computed by equation 3.3. The errors of the Lucas-Kanade/Liu-Shen and Horn-Schunck/Liu-Shen combinations were computed from 8-bit depth images while the errors of the Farnebäck/Liu-Shen combination were computed from 10-bit depth images. All methods featured a pyramidal 2-level pass. For each flow type and OpF method, the plot shows the central 50% of the error distribution (the width of the whiskers), centered around the median (the dot). Color code: — (Purple) for Uniform flow; — (Red) for the flow with a stagnation point; — (Green) for the Rankine vortex with superimposed Uniform flow; — (Orange) for the Rankine vortex; — (Blue) for the planar Poiseuille flow. . . . . | 50 |

|   |    |
|---|----|
| 3.12 JPIV, OpenPIV and QuickLab PIV accuracy vs particle spot sizes and concentration ( $N_i$ ).<br>Results shown are obtained by averaging the SIGs relative error across all flow types at<br>4.0 px maximum displacement. Zero noise data is excluded since it is not realistic. . . . .   | 54 |
| 3.13 QuickLab PIV (PIV) versus dense Lucas-Kanade/Liu-Shen combined Pyramidal L2 box<br>plots by flow type (8-bits) for the absolute ( $\epsilon - SIG$ ) and relative ( $\epsilon R - SIG$ ) SIG errors .  | 54 |
| 3.14 QuickLab PIV (PIV) versus Farnebäck/Liu-Shen combined Pyramidal L2 box plots by flow<br>type (10-bits) for the absolute ( $\epsilon - SIG$ ) and relative ( $\epsilon R - SIG$ ) SIG errors . . . . .  | 55 |
| 3.15 dense Lucas-Kanade versus dense Lucas-Kanade Pyramidal L2 violin plots for the abso-<br>lute ( $\epsilon - SIG$ ) and relative ( $\epsilon R - SIG$ ) SIG errors, all for 8-bit image depth. . . . .   | 56 |
| 3.16 Farnebäck versus Farnebäck Pyramidal L2 violin plots for the absolute ( $\epsilon - SIG$ ) and<br>relative ( $\epsilon R - SIG$ ) SIG errors, all for 12-bit image depth. . . . .  | 57 |
| 3.17 Horn-Schunck versus Horn-Schunck Pyramidal L2 violin plots for the absolute ( $\epsilon - SIG$ )<br>and relative ( $\epsilon R - SIG$ ) SIG errors, all for 8-bit image depth. . . . .   | 58 |
| 4.1 QuickLabPIV-ng—PIV processing steps with options for sub-pixel, validation, and OpF<br>hybridization. . . . .   | 68 |
| 4.2 Unified Modeling Language (UML) simplified class diagram depicting the Job class soft-<br>ware design. . . . .  | 72 |
| 4.3 Unified Modeling Language (UML) simplified component diagram describing the main<br>software components. . . . .  | 74 |
| 4.4 QuickLabPIV-ng GUI window for the PIV/Hybrid PIV sub-pixel interpolation configuration.   | 75 |
| 4.5 QuickLabPIV-ng computational times comparison for Database1. . . . .  | 80 |
| 5.1 Classic PIV processing steps/workflow. . . . .  | 82 |
| 5.2 An illustration of the relative accuracy performance between QuickLabPIV-ng (QL)'s Mod-<br>ern PIV (micro-warping) and OpenPIV's Modern PIV with respect to the absolute and<br>relative errors. Absolute errors are shown in sub-figure <b>(a)</b> , and relative errors are in sub-<br>figure <b>(b)</b> . The vertical black dashed line at the center is the null-advantage line, data to<br>the left indicate a QuickLabPIV-ng advantage, and data to the right favor the OpenPIV<br>method. The rectangles on the left side and right side denote the Q1 quartile and Q3<br>quartile, respectively. The vertical red line denotes the median, or the Q2 quartile, while<br>the right whisker denotes the Q4 quartile. The circles denote outliers that are outside the<br>margin defined by three times the inter-quartile range (IQR), centered around the mean<br>of the distribution, where the IQR is the difference between Q3 and Q1. The advantage is<br>determined by the difference in relative accuracy between the two methods for all the con-<br>sidered parameters. The vertical blue dashed line marks the 5 dB advantage for OpenPIV. | 85 |

- 5.3 The advantage of a last step of sparse the OpF sub-pixel with Liu–Shen combined with Lucas–Kanade over standard sub-pixel interpolation. The vertical blue dashed line marks the 5 dB advantage for the OpF sub-pixel (variant 3). Sub-figure (a) refers to absolute errors, while sub-figure (b) refers to relative errors. For further details, please refer to Figure 5.2 legend. . . . . 86
- 5.4 The advantage of hybrid PIV to standard PIV with warping. The vertical blue dashed line marks the 5 dB advantage for the hybrid PIV mode (variant 1). Sub-figure (a) refers to absolute errors, while sub-figure (b) refers to relative errors. For further details, please refer to Figure 5.2 legend. . . . . 86
- 5.5 An overview of the overall accuracy distribution for the QuickLabPIV-ng method type for all the considered flow types. In sub-figures (a,b), we see the average of the absolute errors in a SIG and the average of the relative errors in a SIG, respectively. The PIV with warping method is PIV with micro-warping and Hongwei Guo’s 1D-1D robust linear regression sub-pixel with a 5 px width and 20 iterations and no velocity vector validation. The PIV Subp OpF method is identical to the PIV warping method, with the exception that Hongwei Guo’s sub-pixel is parameterized with a 3 px width while keeping the same number of iterations, all followed by a final step with sparse optical flow with the Liu–Shen method combined with the Lucas–Kanade method. The hybrid PIV method is identical to the PIV Subp OpF method, with the exception that the final optical flow is computed in dense mode, that is, one velocity vector per image pixel. For further details, please refer to Figure 5.2 legend. . . . . 87
- 5.6 The figures show the absolute error versus actual particle concentration per IA volume ( $N_i F_i F_o$ ) or by image for hybrid PIV ( $N_i F_o$ ), since the latter has no concept of IA. The data pertain to the Rankine vortex with superimposed uniform flow, 4.0 px maximum velocity vector magnitude, and 8-bit depth, with four values of number particle concentration  $N_i$ , namely, 1, 6, 12, and 16 particles per IA volume, and three levels of WGIN noise, 0, 5, and 15 dBW. Sub-figure (a) is the PIV method with micro-warping and Hongwei Guo’s robust linear regression of 1D-1D Gaussian as the sub-pixel method with 5.0 px and 20 iterations; in sub-figure (b), we have the hybrid PIV method with micro-warping employing the same Hongwei Guo’s sub-pixel with a 3.0 px width and 20 iterations, followed by a final step with Lucas–Kanade optical flow; in sub-figure (c), we have the hybrid PIV method with micro-warping employing the same Hongwei Guo’s sub-pixel with a 3.0 px width and a final step with Liu–Shen combined with Lucas–Kanade optical flow. (Legend:  $\times$ —denotes 1 px diameter imaged particle spot size;  $+$ —denotes 2 px diameter of imaged particle spot size;  $\bigcirc$ —denotes 3 px diameter imaged particle spot size;  $\star$ —denotes 6 px diameter imaged particle spot size; Black—denotes 0 dBW WGIN; Green—denotes 5 dBW WGIN; Red—denotes 15 dBW WGIN.) . . . . . 88

- 5.7 All flows compared with respect to relative errors, with a 4.0 px maximum velocity vector magnitude and an 8-bit depth with the average noise level. PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 5.0 px and 20 iterations is shown in sub-figure (a). In sub-figure (b), we have hybrid PIV with micro-warping employing the same Hongwei Guo's sub-pixel and a final step with Lucas–Kanade optical flow. Finally, in sub-figure (c), we have hybrid PIV with micro-warping employing the same Hongwei Guo's sub-pixel and a final step with Liu–Shen combined with Lucas–Kanade optical flow. (Legend:  $\times$ —denotes a Poiseuille flow;  $+$ —denotes the Rankine vortex;  $\bigcirc$ —denotes the Rankine vortex with a superimposed uniform flow;  $\star$ —denotes an inviscid flow with a stagnation point;  $\diamond$ —denotes a uniform flow; Red—denotes 1 particle per IA volume; Black—denotes 6 particles per IA volume; Green—denotes 12 particles per IA volume; Blue—denotes 16 particles per IA volume.) . . . . . 88
- 5.8 Spatial distributions of absolute and relative errors represented through Figures (x1) and Figures (x2), respectively, where x is either a, b, or c. These figures are for the Poiseuille flow, with a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures (ax), where x is either 1 or 2, present velocimetry data obtained by PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 5.0 px and 20 iterations. Figures (bx), where x is either 1 or 2, present velocimetry data obtained by PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 3.0 px and 20 iterations followed by a last step with sparse optical flow with Liu–Shen combined with Lucas–Kanade. Figures (cx), where x is either 1 or 2, present velocimetry data obtained by hybrid PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 3.0 px and 20 iterations followed by a last step with dense optical flow with Liu–Shen combined with Lucas–Kanade. The data shown are the mean relative or mean absolute errors of a SIG for each IA (for PIV methods) or each pixel (for hybrid PIV). . . . . 90
- 5.9 Spatial distribution of absolute and relative errors represented through Figures (x1) and Figures (x2), respectively, where x is either a, b, or c. These figures are for the Rankine vortex, with a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures (ax), (bx) and (cx), where x is either 1 or 2, represent the same parameterizations as in Figure 5.8. . . . . 91
- 5.10 The spatial distribution of relative errors represented through Figures a, b, or c. The figures for the absolute errors are not shown, as they are mostly identical to their equivalents in Figure 5.9. These figures are for the Rankine vortex with a superimposed uniform flow, having a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures (a–c) have the same method configurations as (a2–c2) in Figure 5.8. . . . . 91

|  |     |
|--|-----|
| 5.11 Spatial distribution of relative errors represented through Figures <b>a</b> , <b>b</b> , or <b>c</b> . Absolute errors are not shown since they are essentially low everywhere. These figures are for the inviscid flow with a stagnation point, having a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures ( <b>a–c</b> ) have the same method configurations as ( <b>a2–c2</b> ) in Figure 5.8. . . . . | 92  |
| 5.12 Spatial distribution of absolute errors represented through Figures <b>a</b> , <b>b</b> , or <b>c</b> . Relative errors are not shown since they are essentially negligible everywhere. These figures are for the uniform flow, with a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures ( <b>a–c</b> ) have the same characteristics as their equivalents ( <b>a1–c1</b> ) in Figure 5.8 . . . . .        | 93  |
| 5.13 The top row shows median background images after intensity normalization. The bottom row shows sample PIV images from the image database. The flow is from left to right hand sides. ( <b>a</b> ) Cylinder database (background); ( <b>b</b> ) Boundary layer db (background); ( <b>c</b> ) Plunging flow database (background); ( <b>d</b> ) Cylinder database (PIV image); ( <b>e</b> ) Boundary layer db (PIV image); ( <b>f</b> ) Plunging flow database (PIV image). . . . .                 | 96  |
| 5.14 The power spectral density (PSD) in the cylinder wake as a function of the non-dimensional frequency at the normalized location ( $X = 0.58d, Y = 2.28d$ ), where $d$ is the cylinder diameter, and the referential origin is the tip point on the lee side of the cylinder. The vertical dashed lines represent the Strouhal number of 0.21. . . . .   | 97  |
| 5.15 Cylinder database—the power spectral density (PSD) function for a spatial series of longitudinal velocity components at $y = 7.626d$ , the region far from the cylinder wake. The center of the cylinder is located at $x = 1.540d, y = 5.322d$ . . . . .   | 98  |
| 5.16 The boundary layer database—the auto-correlation function of the spatial series of longitudinal velocity components at at 7% of the flow depth. . . . .   | 99  |
| 5.17 The boundary layer database—the power spectral density (PSD) function of the spatial series of longitudinal velocity components at at 7% of the flow depth. . . . .   | 99  |
| 5.18 The plunging flow database—the auto-correlation function for the bed-normal velocity component for a region with similar mean velocities. . . . .   | 101 |
| 5.19 The plunging flow database—the power spectral density (PSD) function for the bed-normal velocity component in a region with similar mean velocities. . . . .  | 101 |
| A.1 Portable PIV LED lighting unit. . . . .  | 114 |
| A.2 Portable PIV LED lighting unit - LCD screen. . . . .   | 114 |
| A.3 Portable PIV LED lighting unit in operation. . . . .   | 115 |
| A.4 Sample PIV image from a LED PIV illuminated setup. . . . .   | 116 |
| A.5 LED PIV validation - Mean velocity norm with superimposed sparse mean velocity vectors. . . . .  | 117 |
| A.6 LED PIV validation - Longitudinal velocity map. . . . .  | 117 |
| A.7 LED PIV validation - Normal velocity map. . . . .  | 118 |
| A.8 LED PIV validation - Reynolds proxy for the longitudinal component. . . . .  | 118 |



|   |     |
|---|-----|
| A.9 LED PIV validation - Reynolds proxy for the normal component. . . . . | 119 |
| A.10 LED PIV validation - Reynolds proxy for the cross component. . . . . | 119 |



# Nomenclature

## Greek symbols

$\rho$  Density

## Roman symbols

$p$  Pressure

$\mathbf{u}$  Velocity vector

$u, v, w$  Velocity Cartesian components

## Subscripts

$i, j, k$  Computational indexes

$x, y, z$  Cartesian components

## Superscripts

$*$  Adjoint

$T$  Transpose



# Glossary

|              |  |
|--------------|--|
| <b>CPU</b>   | Central Processing Unit                  |
| <b>FFT</b>   | Fast Fourier Transform                   |
| <b>GpGPU</b> | Generic purpose Graphics Processing Unit |
| <b>GUI</b>   | Graphical User Interface                 |
| <b>HPC</b>   | High Performance Computing               |
| <b>IA</b>    | Interrogation Area                       |
| <b>JAR</b>   | Java Archive                             |
| <b>JVM</b>   | Java Virtual Machine                     |
| <b>LED</b>   | Light Emitting Diode                     |
| <b>NVMe</b>  | Non-Volatile Memory                      |
| <b>OpF</b>   | Optical Flow                             |
| <b>PIV</b>   | Particle Imaging Velocimetry             |
| <b>PSD</b>   | Power Spectrum Density                   |
| <b>RAM</b>   | Random Access Memory                     |
| <b>SIG</b>   | Statistical Image Group                  |
| <b>SIMT</b>  | Single Instruction Multiple Threading    |
| <b>UML</b>   | Unified Modeling Language                |
| <b>VRAM</b>  | Video Random Access Memory               |
| <b>WGIN</b>  | White Gaussian Image Noise               |



# Chapter 1

## Introduction

### 1.1 Motivation and research questions

Velocity fields are a key parameter in various scientific disciplines, providing insights into the dynamics of moving objects and fluids. In physics, velocity fields are essential for understanding motion and predicting future states of physical systems. They are used to calculate other physical quantities like acceleration, force, and momentum, which are foundational concepts in mechanics. Also, the engineering field relies in velocity fields to design and optimize systems and structures, such as vehicles, buildings, and bridges, ensuring they can withstand dynamic forces like wind and water flow. In sports science, understanding the aerodynamics of helmets, suits and bicycles, or even a platoon, is commonly done for performance analysis and improvement, also relying on velocity field analysis. In overall, velocity fields are indispensable for the analysis and understanding of dynamic systems across scientific domains. They provide a quantitative measure of how fast and in what direction an object or fluid element is moving, being relevant for research, development, and practical applications in science and engineering.

Particle Image Velocimetry (PIV) belongs to the optical measurement techniques, enabling the extraction of velocity fields with reduced intrusiveness. This method necessitates optical access for a high-speed camera, a collimated light sheet produced by a laser or LED, and tracer particles dispersed within the medium under investigation. The high-speed camera captures the light reflected from the particles as they traverse the illuminated light sheet, which corresponds to the flow field imposed on the medium. The resultant PIV images exhibit high contrast, typically monochromatic, with a dark background and bright spots representing the light reflections or scattering from the imaged particles. Subsequent image processing employs a PIV algorithm, which commonly adopts the principle of dividing the image into uniformly spaced, squared regions termed Interrogation Areas (IAs). These IAs are then cross-correlated between successive frames—or between image pairs in the case of double-pulsed light sources—to deduce the flow dynamics. The PIV software computes a velocity field, assigning a two-dimensional velocity vector to each IA. This capability has established PIV as an essential tool across diverse fields, encompassing both industrial applications and academic research. PIV facilitates com-

prehensive flow visualization, enabling researchers to investigate complex flow phenomena with minimal or no perturbation to the flow's natural state. Through PIV, detailed insights into fluid mechanics are obtained, enhancing our understanding and enabling the optimization of various processes influenced by fluid dynamics.

Another class of methods that has long been used to determine the apparent motion of image objects between two consecutive frames is the Optical Flow (OpF) class of methods. With such methods, the apparent movement is either caused by the object displacement, the camera displacement, or by lighting condition changes, resulting in a 2D vector field of velocities, similar to what is obtained in PIV. Optical flow methods have however received little attention for fluid mechanics applications, with some attempts mostly based on the Horn-Schunck algorithm, or derivations of it, that evolved to computationally intensive methods. Only a few other OpF methods, other than Horn-Schunck, have been validated for fluid mechanics, from which we highlight Lucas-Kanade derived methods, like FOLKI, or the Liu-Shen Physics Based Optical Flow method, where the latter uses the output of the Horn-Schunck methods as the initial motion estimate for the Liu-Shen method. Many more OpF methods exist, however and are intensively used in the Computer Vision field, which may or may not be suitable for fluid mechanics applications. Among them, we have Farnebäck's two frame based estimation method and dense Lucas-Kanade, where dense indicates one velocity vector per image pixel. In fact, most OpF methods tend to be more computational efficient than PIV, with some works pointing to real time or near-real time processing times. Also, most OpF methods contrary to PIV provide one velocity vector per each image pixel, instead of one velocity vector per IA.

In the last few years some experimentation was carried out for hybridizing PIV algorithms with OpF methods, which have shown promising results with increased accuracy and with dense vector field capabilities, but not by any means exhaustive, given the sheer amount of OpF methods available. Thus, it can be relevant to find if there are other possible methods with better potential for hybridization, that can improve both accuracy and spatial resolution. In fact, to our knowledge there is a lack of studies regarding the performance of OpF and hybrid PIV methods, with respect to the recovery of turbulent scales and their spatial resolution for  $2^{nd}$  order structure function, auto correlation and power spectral density.

The main focus of this thesis is to assess if such OpF techniques can be utilized either in combination with PIV, or not, in order to increase the accuracy and spatial density of the velocity fields in a way that allows better characterization of turbulent flows in natural and built environments, including in laboratory environments. Besides that main focus we are also interested in OpF arrangements that are lightweight so that they can operate on edge computing devices with real-time or near real-time performance in order to enable river flow and flash floods monitoring for rescue and prevention of such natural disasters.

## 1.2 Objectives and Deliverables

The main objective of this thesis is the in-depth study of efficient PIV and Optical Flow methods (OpF) for particle imaging velocimetry applications, that allow an increase of the spatial resolution with



respect to standard PIV methods keeping near-real-time performance, so that a solution can be derived and deployed into edge computing hardware. The second objective of this work is to develop a portable low-cost PIV LED lighting unit to facilitate PIV courses to undergraduate students.

The main steps entailed:

- Designing an open-source synthetic PIV image database generator intended for the benchmark (with respect to accuracy) of PIV and Optical Flow algorithms;
- Analyzing the state of the art and selecting a set of candidate set of OpF methods (Horn-Schunck, Lucas-Kanade, Färneback and Liu-Shen Physics based optical flow) for evaluation against the synthetic database, looking at spatial error locations, as well as average and absolute errors;
- Elaborating a PIV/ hybrid PIV open-source software for benchmarking hybrid PIV with near real-time performance and evaluating OpF methods with it for large image datasets. Essentially a software that can be extended for several OpF methods with a set of higher level APIs, targeting parallel CPU and massively parallel GpGPU computation;
- Evaluating one of the best performing algorithm combinations in a hybrid PIV configuration, thus including Lucas-Kanade combined with Liu-Shen, as well as hybrid PIV with Lucas-Kanade. Both methods analyzed in terms of accuracy with respect to its spatial error location and their performance regarding turbulent scales recoveries, the latter with a real set of PIV databases having different image quality or characteristics.
- Building and testing a LED PIV lighting unit.

All these aspects are fundamental for real-time implementation on custom hardware, contributing to the goals described, either directly, by providing ready to use software, or indirectly, by paving the way for future works.

## **1.3 Impact**

### **1.3.1 Potential for technology transfer**

Floods and flash floods severely impact human lives, the infrastructures and economy. The climate change, seems to aggravate this scenario by making severe flash floods more common. Faster and more accurate approaches for prediction of flood progression are needed to empower prevention and to reduce losses. One such approach, consists on data assimilation techniques to enhance the quality of the estimates of a flood progression. Advanced prediction allows to plan evacuations and rescue operations before a flood reaches critical proportions. IoT (Internet of Things) sensor networks can sense the environment and send data to be processed on the data assimilation server. Compact and cost effective edge devices must make onboard computation of real-time free-surface velocimetry, so that instead of transmitting the images for central processing, they can send the velocimetry data to largely alleviate both the servers load and the required data transmission capabilities. These devices could be installed

into fixed poles for permanent monitoring, or in portable equipment for measurements on inaccessible or otherwise non-permanently monitored areas. Since edge devices have limited computation capabilities and stronger power consumption limits a lightweight OpF/hybrid PIV method that is suitable for fluid mechanics velocimetry is a needed first step.

### **1.3.2 Benefits to research institutions**

Currently the costs of commercial PIV software and equipment are mostly prohibitive for teaching purposes, making PIV inaccessible to undergraduate students, which effectively delays the learning of modern velocimetry methods. It is costly to equip a laboratory with multiple PIV units, and making the software easily accessible to students, off the lab. Open-source PIV software does exist, but it is either slow, not easy to use and it does not contemplate the hardware, which also incurs in significant costs.

Other possible use cases include high end research on the PIV method itself, since the most performant PIV software is typically closed source, and the existing open-source ones are not designed with extensibility in mind. It is not trivial to test new PIV methods over large datasets.

In this thesis we have developed a novel open-source accurate and extensible software with Graphical User Interface (GUI) that explores a novel hybrid PIV-OpF method with Liu-Shen combined with Lucas-Kanade OpF. This novel hybrid PIV-OpF method is assessed both with synthetic PIV image databases and with real laboratory PIV image databases of turbulent flows, leading to increased spatial velocity field resolution under certain conditions. A novel Liu-Shen combined with Farnebäck OpF method is also presented and studied. In addition we propose a low cost PIV LED lighting unit to facilitate PIV adoption by undergraduate students, not only reducing device cost, but also removing the considerable danger of high-power LASERs.

## **1.4 Thesis Outline**

This thesis is built as a collection of published journal articles that entail a journey that starts by studying existing PIV and OpF methods in terms of their accuracy and ends with a novel hybrid PIV-OpF method incorporated into a novel HPC open-source software with GUI, named QuickLabPIV-ng. In between, it considers novel OpF method combinations like the Liu-Shen combined with Lucas-Kanade and the Liu-Shen combined with Farnebäck OpF methods. During this journey the presented goals and deliverables are incrementally achieved. The thesis is outlined in four essential chapters that we described next.

In the process of assessing PIV and OpF methods with respect to their accuracy based performance, we found that there was a lack of open-source, or otherwise easily accessible synthetic PIV image generators with supporting ground-truth data. This void, allowed for the creation of such a tool, seeking easy parameterization and easy extension with more flow types, that culminated in a work that was published in an open-access journal, [1]. This published work is presented in the first of the four main chapters and paves the way for the remaining work in this thesis.

The second main chapter leverages the PIV image generator open-source tool developed in the first main chapter to assess the performance of existing and novel OpF methods applied to the hydraulics domain, work that corresponds to the published journal article [2]. In this chapter a synthetic PIV database of images crafted with unique characteristics including image characteristics, particle sizes, particle concentrations, minimum and maximum velocities, out of plane movement, velocity gradients and flow types of rotational based, distortional based and flows with singularities are considered. These special crafted image characteristics are utilized to assess the method's compatibility with PIV and the method's accuracy based performance with respect to the theoretical solutions of the velocity fields of such flows. Two well known OpF methods in the hydraulics domain are considered, namely the Horn-Schunck and the Physics based Optical Flow methods. Another two methods well known in the Computer Vision field, but less considered for fluid mechanics applications, consist on the dense Lucas-Kanade and the Farnebäck methods, especially the latter. Novel combinations of the Liu-Shen method with the dense Lucas-Kanade and the Liu-Shen combination with the Farnebäck method are also considered.

The third main chapter leverages the novel Liu-Shen combined with Lucas-Kanade OpF method, studied in the preceding chapter, to create a novel hybrid PIV-OpF method that combines PIV with the said OpF method and that leads to a novel open-source HPC software, named QuickLabPIV-ng. This software is presented in the first part of the journal article [3] which has been split into two chapters in this thesis, so that it could be extended with a summary of the software design and architecture, as well sample execution times and configurations.

Finally, in the fourth and last main chapter, the new software is utilized to assess the performance of the method across large image datasets that include a set of real PIV laboratory image databases of turbulent flows that are utilized to verify the method's ability to extend the turbulence range to smaller scales, when compared to standard PIV techniques. The PSD (Power Spectral Density) and Autocorrelation functions are employed in the verification process of expected turbulence characteristics. The hybrid method is also evaluated under the same synthetic image database following a similar approach as employed in the OpF methods to assess its overall accuracy. Relative performance between this novel OpF method and PIV are studied and the relative and absolute spatial error distributions of the PIV and OpF methods are presented. This chapter refers to the second part of the journal article [3].

The conclusions and future work are presented in the last chapter. A small appendix that presents the PIV LED lighting unit build and testing, is also included.

## 1.5 Related works

In this section we present all the thesis related works that were carried out.

Journal publications:

- L. Mendes, A. Bernardino, and R. M. Ferreira. piv-image-generator: An image generating software package for planar piv and optical flow benchmarking. *SoftwareX*, 12:100537, 2020. ISSN 2352-7110. doi: <https://doi.org/10.1016/j.softx.2020.100537>. URL <https://www.sciencedirect.com/science/article/pii/S2352711020300339>.

- L.P.N. Mendes; A.M.C. Ricardo; A.J.M. Bernardino; R.M.L. Ferreira, A comparative study of optical flow methods for fluid mechanics. Exp. Fluids **2021**, 63, 7. doi: <https://doi.org/10.1007/s00348-021-03357-7>.
- L.P.N. Mendes; A.M.C. Ricardo; A.J.M. Bernardino; R.M.L. Ferreira, A Hybrid PIV/Optical Flow Method for Incompressible Turbulent Flows. Water **2024**, 16, 1021. doi: <https://doi.org/10.3390/w16071021>.

#### Conference papers:

- L.P.N. Mendes; A.J.M. Bernardino; R.M.L. Ferreira, A customizable open-source PIV software platform. 2<sup>nd</sup> HydroSenSoft 2019 - International Symposium and Exhibition on Hydro-Environment Sensors and Software 2019, **2019** (Spain, March 2019). ISBN 978-90-824846-4-9
- L.P.N. Mendes; A.J.M. Bernardino; R.M.L. Ferreira, Comparison of PIV and Optical flow for river flow applications. In River Flow 2020: Proceedings of the 10th Conference on Fluvial Hydraulics, **2020** (Delft, Netherlands, 7-10 July 2020). CRC Press, Eds. Wim Uijttewaalt et al. ISBN: 9780367627737
- L.P.N. Mendes; A.J.M. Bernardino; R.M.L. Ferreira, An assessment of optical flow methods for river flow velocimetry. 15<sup>th</sup> Congresso da Água, **2021**, March, Lisbon, Portugal.
- L.P.N. Mendes; A.J.M. Bernardino; R.M.L. Ferreira, Optical flow methods for tracer-based imaging techniques in fluid mechanics. In Proceedings of the 20th International Symposium of Laser and Imaging Techniques to Fluid Mechanics **2022** (Lisbon, Portugal, July, 14<sup>th</sup>) pp.1810-1820 ISBN: 978-989-53637-0-4.
- L.P.N. Mendes; A.M.C. Ricardo; A.J.M. Bernardino; R.M.L. Ferreira, PIV without cross-correlation: an assessment of optical flow methods. In Abstract Book: 6th IAHR Europe Congress, **2021**, pp. 147, 148, Warsaw, Poland.

#### Other disseminations:

- D. Valero et al., Pathways towards democratization of hydro-environment observations and data. Pathways towards democratization of hydro-environment observations and data, IAHR White Paper Series **2021**, **1**. [https://static.iahr.org/library/AnythingElse/WhitePaper/2021\\_1\\_Democratization\\_Hydro\\_Environment\\_Observations\\_Data.pdf](https://static.iahr.org/library/AnythingElse/WhitePaper/2021_1_Democratization_Hydro_Environment_Observations_Data.pdf)

#### Software releases:

- **piv-image-generator**: An image generating software package for planar PIV and Optical Flow benchmarking.
- **QuickLabPlv-ng**: A Particle Imaging Velocimetry (PIV) and dense hybrid PIV-OpF Software Laboratory with GpGPU processing.

#### Hardware devices:

- PIV-LED Portable Light Unit.

## Chapter 2

# An image generating tool for planar PIV and Optical Flow benchmarking

Article published in the SoftwareX journal with DOI: <https://doi.org/10.1016/j.softx.2020.100537>

Received 7 February 2020, Revised 4 June 2020, Accepted 4 June 2020, Available online 9 June 2020, Version of Record 9 June 2020.

### 2.1 Abstract

We describe a generator of synthetic images of tracers in turbulent flows to benchmark Particle Image Velocimetry and Optical Flow algorithms. Generated flows include: uniform, shear flows, inviscid stagnation point flows, and decaying and Rankine vortices. Image control parameters include particle and illumination characteristics, noise level and image bit depth, among others. The tool is unique in the sense that image characteristics and error sources are fully parameterized and that it provides zero-uncertainty ground truth, a combination of features never before conjoined in synthetic particle image generators. Examples of application are included in the text.

Keywords: PIV Optical Flow Hydraulics Fluid Mechanics Velocimetry

## Required Metadata

### Current code version

| Nr. | Code metadata description                                       | Please fill in this column  |
|-----|---|---|
| C1  | Current code version  | v1.1.2  |
| C2  | Permanent link to code/repository used for this code version    | <a href="https://git.qoto.org/CoreRasurae/piv-image-generator/tree/v1.1.2">https : //git.qoto.org/CoreRasurae/piv-image-generator/tree/v1.1.2</a> |
| C3  | Code Ocean compute capsule                                      |   |
| C4  | Legal Code License  | GNU General Public License v2   |
| C5  | Code versioning system used                                     | git   |
| C6  | Software code languages, tools, and services used               | MATLAB  |
| C7  | Compilation requirements, operating environments & dependencies | MATLAB for Linux, OS X or Windows   |
| C8  | If available Link to developer documentation/manual             | A Technical manual PDF is provided in the submission  |
| C9  | Support email for questions                                     | luis.mendes@tecnico.ulisboa.pt  |

Table 2.1: Code metadata (mandatory)

## 2.2 Motivation and significance

Planar Particle Image Velocimetry (PIV) is an image processing method supported by cross correlation, which is able to extract velocity vectors with two components in a two dimensional grid from a pair of tracer images taken with a fixed time interval. Tracer images are black and white images of bright spots against a dark background resulting from light reflected from tracer particles when moving across an illuminated light sheet. These are artificially inserted in the medium of interest (e.g. fluid, air, etc). Are in suspension in the medium, are small and ideally have a density close to that of the medium, so that their movement is highly related with the flow or the medium. The grid is normally a matrix of contiguous regular elements, called interrogation areas (IAs). A two component velocity vector is obtained for each IA, for each image pair.

PIV is widely used in industry applications and in scientific explorations, see e.g. [2]. Example applications include fluid mechanics, in general [1], aerodynamics, e.g. aircraft and airfoil profile design [10], characterization of building and bridge aerodynamic actions [12], sports science [5], combustion, e.g. spray optimization in internal engines [20], hydraulics, e.g. environmental flows [18] or bio fluid mechanics e.g. vascular mechanics [4]. Optical Flow (OF) methods are widely used in the robotics field [7], computer vision [13], autonomous vehicles [8], but also, for replacing or complementing PIV cross-correlation usage cases [9].

Databases of tracer based images that include the ground truth to validate velocimetry instrumentation in fluid applications are not common. The few relevant examples, e.g. [1, 2], have provided valuable means to assess the quality of PIV and Optical Flow algorithms [11]. However, to the best of our knowledge, we find that there are no databases or tools to generate images that simultaneously: i) include rotational and distortional flows and combinations of both as well as flows with singularities; ii) provide

the ground truth based on theoretical solutions and thus with no uncertainty or, at least, low and quantifiable uncertainty; iii) include the main causes for error in PIV measurements as parameters that can be manipulated by the user when generating the images (the latter include size of particles, out-of-plane movement, white noise and particle-related noise, tracer density, thickness of the light sheet, image bit depth, number of repetitions with same parameters - for statistical relevance); iv) allow for the precise specification of the number of tracer particles in the image and the number of particles whose illumination is below the threshold (and are considered noise), the reference maximum displacement in a ROI (region of interest), and the stochastic control of out-of-plane motion.

We aim at bridging this knowledge gap by creating a tool that generates such databases from a set of user-defined relevant control parameters including tracer image size, particle concentration, out-of-plane motion, white noise level or pixel bit depth.

The images are relative to different user-defined flow types. The software can be easily extended to support additional types, should users identify extra needs. Besides generating test images, the ground truth velocity is also exported by the tool. A single MATLAB data file is created that contains validation velocities for PIV methods (generated at the centre of IAs) as well as validation velocities for dense Optical Flow (OF) methods (generated at the centres of pixels).

The software can be used to provide test images pairs and reference data for performing comparison and benchmarking of PIV and OF methods regarding their overall and spatially specific accuracy, for the different control parameters that the image generator supports. We provide, in the software repository, an example for sample tracer based image generation [14].

The paper is structured around four other sections. Section "Software description" provides the software fundamentals and gives a general overview. Section "Illustrative Examples" provides some examples for software use cases. Section "Impact" sketches how the software contributed to research developments and its expected near future influence for research, as well as, for the society. Closing the paper, section "Conclusions" presents a final discussion and lines out possible future work.

## **2.3 Software description**

### **2.3.1 General description**

The software processing phases can be summarized in the following seven steps:

- 1 The user defines the image generation parameters
- 2 A planar flow field is generated according to the user preferences
- 3 Particles are randomly placed, including the lateral out of plane initial position, in the first image
- 4 Particles are displaced according to the planar flow field in the second image, and according to the lateral (out of plane) movement
- 5 Images are rendered

6 Noise perturbations are superimposed to the image

7 Images are exported to file in TIF format, along with validation data file in MATLAB format

### 2.3.2 Flow types

Tracer images are black and white images of bright spots against a dark background. Resulting from light reflected from tracer particles when moving across an illuminated light sheet. These are artificially inserted in the medium of interest (e.g. fluid, air, etc). Are in suspension in the medium, are small and ideally have a density close to that of the medium so that their movement is highly related with the flow or the medium.

This tool includes the following flows:

- 1 uniform flow;
- 2 Poiseuille flow;
- 3 simple shear flow;
- 4 inviscid stagnation point flow [3];
- 5 Rankine vortex;
- 6 Rankine vortex with superimposed uniform flow, and
- 7 decaying vortex (similar to a Lamb-Oseen vortex, as proposed by [2]).

Displacement fields are found by analytically integrating the corresponding velocity field in all but one case, the Rankine vortex with superimposed uniform flow. In the latter case, a numerical integration with a 4<sup>th</sup>-order Runge-Kutta method is employed. The uncertainty in particle position is zero in all cases but #6; in the case of flow #6, the error in particle positioning can be neglected as is  $O(\Delta t)^3$  and the time interval,  $\Delta t$ , between images, is small for both PIV and OF applications.

Table 2.2 details the equations that describe the above flows and corresponding displacement fields, as implemented in the tool. In these equations,  $x$  and  $y$  are Cartesian orthogonal spatial coordinates specifying a plane,  $x_0$  and  $y_0$  are initial positions,  $r$  is the radial polar coordinate,  $\theta$  is the azimuthal polar coordinate,  $\theta_0$  is an initial azimuthal position,  $t$  is the time elapsed since an initial instant,  $x_c$  and  $y_c$  are the coordinates of the zero-velocity locus, in the case of the Shear Flow, or the coordinates of the stagnation point, for the inviscid stagnation point flow, or the centre of the vortex, for both Rankine and Decaying vortices;  $y_{max}$  and  $y_{min}$  are the coordinates of the upper and lower bounds of the flux tube within which the Poiseuille flow takes place;  $R$  is the radius for which maximum azimuthal velocity is attained in the decaying and Rankine vortex flows,  $\Gamma$  is the circulation of rotational core, in the Rankine vortex,  $m$  is a reference velocity,  $M$  is a scaling parameter that allows for the specification of the desired maximum relative displacement (a percentage of the IA size, always less than 25%). For the Rankine Vortex and Rankine with superimposed Uniform flow cases  $\Gamma = m2\pi R$ . For the decaying vortex,  $\Gamma = m2\pi R$  and  $\alpha = 1.25643$  are parameters that express the vortex core circulation, as specified by [2].



This flow field parameterization is entirely independent of PIV or Optical flow image variables, such as the size of the IA, as  $m$  is an actual velocity, expressed in m/s. In the case of PIV applications, however, it is intended that the maximum displacement is a fraction of the size of the PIV IA. This is achieved by specifying appropriate values of  $M$  to transform the velocity from physical coordinates to image coordinates per unit time. In the parabolic flow, for instance, the scaling parameter  $M$  is:

$$M = (\frac{y_{max} - y_{min}}{2} - y_{min})(\frac{y_{max} - y_{min}}{2} - y_{max}).$$

The value of the time stamp of the second image,  $t$ , is computed as  $t = \frac{M}{m} D_{max}$  where  $D_{max}$  is the maximum displacement (in pixels). In the case of PIV applications,  $D_{max} = p I_A$  where  $p$  denotes the percentage of the size of the PIV interrogation area  $I_A$  (in px).

The set of flows included in this tool is adequate and sufficient for the purpose of testing planar PIV and Optical Flow algorithms. In continuum mechanics, all velocity gradients can be decomposed into a distortional component and a rotational component. The set of flows included in this tool represent both distortional and rotational flows as well as singularities and simple zero gradient flows. The uniform flow, for which the velocity gradient is zero, is the simplest case. The shear and Poiseuille flows are simultaneously rotational and distortional, the decaying vortex and the core of the Rankine vortex are purely rotational, and the axisymmetric stagnation point flow and the outer ring of the Rankine vortex are purely distortional. Additionally, the superposition of a uniform flow to a Rankine vortex is included as an example of a flow for which the streamlines cannot be used to base criteria to detect rotational motion.

### 2.3.3 Software Architecture

The main software functionality is to generate synthetic PIV images that mimic true PIV images obtained in laboratory environments, while having zero-uncertainty ground truth. The images and underlying flow fields are generated according to the flow diagram presented in Figure 2.1.

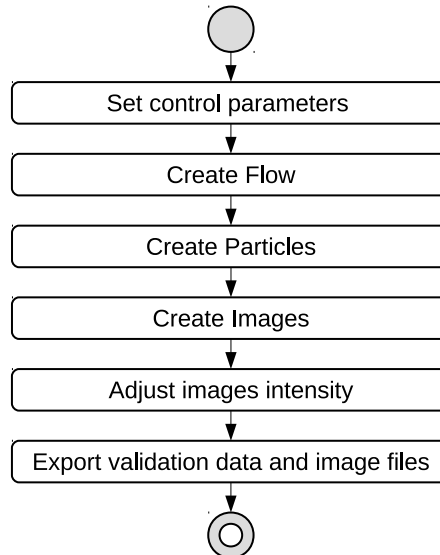


Table 2.2: Flow types, velocity flow fields and displacement equations.

| Flow type                 | Displacements in the flow field   | Velocity flow field   |
|---------------------------|---|---|
| Uniform                   | $\begin{cases} x_1 = \frac{m}{M}t + x_0 \\ y_1 = y_0 \end{cases}$   | $\begin{cases} u = \frac{m}{M} \\ v = 0 \end{cases}$  |
| Poiseuille                | $\begin{cases} x_1 = \frac{m}{M}(y_0 - y_{min})(y_0 - y_{max})t + x_0 \\ y_1 = y_0 \end{cases}$   | $\begin{cases} u = \frac{m}{M}(y - y_{min})(y - y_{max}) \\ v = 0 \end{cases}$  |
| Stagnation                | $\begin{cases} x_1 = e^{\frac{m}{M}x}t(x_0 - x_c) + x_c \\ y_1 = e^{-\frac{m}{M}y}t(y_0 - y_c) + y_c \end{cases}$   | $\begin{cases} u = \frac{m}{M}x(x - x_c) \\ v = -\frac{m}{M}y(y - y_c) \end{cases}$   |
| Simple shear              | $\begin{cases} x_1 = \frac{m}{M}y_0t + x_0 \\ y_1 = y_0 \end{cases}$  | $\begin{cases} u = \frac{m}{M}y \\ v = 0 \end{cases}$   |
| Shear rotated by $\theta$ | $\begin{cases} x_1 = \frac{m}{M}((y_0 - y_c)\cos(\theta) + (x_0 - x_c)\sin(\theta))\cos(\theta)t + x_0 \\ y_1 = -\frac{m}{M}((y_0 - y_c)\cos(\theta) + (x_0 - x_c)\sin(\theta))\sin(\theta)t + y_0 \end{cases}$   | $\begin{cases} u = \frac{m}{M}((y - y_c)\cos(\theta) + (x - x_c)\sin(\theta))\cos(\theta) \\ v = -\frac{m}{M}((y - y_c)\cos(\theta) + (x - x_c)\sin(\theta))\sin(\theta) \end{cases}$   |
| Rankine                   | $\begin{cases} x_1 = r\cos(\frac{\Gamma}{2\pi R}\frac{1}{MR}t + \theta_0) \\ y_1 = r\sin(\frac{\Gamma}{2\pi R}\frac{1}{MR}t + \theta_0) \end{cases}, r \leq R$<br>$\begin{cases} x_1 = r\cos(\frac{\Gamma}{2\pi R}\frac{R}{Mr^2}t + \theta_0) \\ y_1 = r\sin(\frac{\Gamma}{2\pi R}\frac{R}{Mr^2}t + \theta_0) \end{cases}, r > R$ | $u_\theta(r) = \begin{cases} \frac{\Gamma}{2\pi R}\frac{r}{MR} & , r \leq R \\ \frac{\Gamma}{2\pi R}\frac{R}{Mr} & , r > R \end{cases}$   |
| Decaying                  | $\begin{cases} x_1 = r\cos(\omega_v(t, \theta_0, r)) \\ y_1 = r\sin(\omega_v(t, \theta_0, r)) \\ \omega_v(t, \theta_0, r) = \frac{\Gamma}{2\pi R}(\frac{1}{r} + \frac{1}{2\alpha}\frac{R}{r^2}) \times \\ \times \frac{1}{M} \left(1 - e^{(-\alpha\frac{r^2}{R^2})}\right) t + \theta_0 \end{cases}$                              | $u_\theta(r) = \frac{\Gamma}{2\pi R} \left(1 + \frac{1}{2\alpha}\frac{R}{r}\right) \times \frac{1}{M} \left(1 - e^{(-\alpha\frac{r^2}{R^2})}\right)$  |
| Uniform and Rankine       | Numerically solved by $4^{th}$ order Runge-Kutta  | $\begin{cases} u = -\frac{\Gamma}{2\pi R}\frac{y}{MR} + u_0 \\ v = \frac{\Gamma}{2\pi R}\frac{x}{MR} + v_0 \end{cases}, r \leq R$<br>$\begin{cases} u = -\frac{\Gamma}{2\pi R}R\frac{y}{M(x^2+y^2)} + u_0 \\ v = \frac{\Gamma}{2\pi R}R\frac{x}{M(x^2+y^2)} + v_0 \end{cases}, r > R$ |

Figure 2.1: Flowchart diagram of PIV image generator tool

The user defines the control parameters for the synthetic generator in the application entry files. The tool then combines them with pre-defined expressions or constants to derive the configuration input parameters for the supporting library. Further details about those internal expressions and constants can be found in the technical manual. The Create Flow step then instantiates the proper flow model and generates the flow field from it. With the instantiated flow field and control parameters, a number of particles are generated according to the  $N_i$  concentration and uniformly randomly distributed across each IA, both in-plane wise, as out-of-plane wise. Displacements are then computed for the out-of-plane movement. Finally, in Create Images step, particles are rendered into an image pair, according to peak intensity, image resolution and flow field displacement. White Gaussian Image Noise (WGIN) is superimposed to the images. Peak image intensity is either clipped, or normalized, depending on user preferences. Finally the image bit depth resolution is enforced in `AdjustImagesIntensity` function. The last step then exports the validation data into a MAT file along with the image pair, that is exported in TIF format.

For further implementation details the reader may consult the software technical manual and the source code itself.

### 2.3.4 Image parameterization and generation

The software generates pairs of images of tracers illuminated by a Gaussian laser sheet. All particles are characterized by the coordinates of their centres in the 3D Cartesian space. The projection of these coordinates in the  $x, y$  plane is randomly generated in the first image. The lateral coordinate, i.e. the out-of-plane motion, is also randomly generated for the first image. The user decides how many particles should be visible in each interrogation area by choosing the value of  $N_i$ , the number particle concentration (number of illuminated particles per PIV interrogation area). The size of the interrogation area,  $I_A$ , must be previously chosen. The total number of generated particles is however larger, with some particles placed beyond the threshold  $I_0/e^2$ , where  $e$  is the Euler's number, of the laser sheet intensity, where  $I_0$  is the maximum intensity. The thickness of the laser sheet (out-of-plane distance encompassing the region where laser intensity is larger than  $I_0/e^2$ ), its standard variation and the maximum intensity are susceptible to be modified by the user.

For each flow type identified in section 2.3.2, the centres of each particle in the  $x, y$  coordinates are displaced as per the equations shown in Table 2.2. So, in the initial image, the number of illuminated particles in each IA is exactly  $N_i$ , but that may not be the case for the second image. The lateral coordinate, is randomly generated to simulate the effect of turbulence. The turbulence intensity is defined by the user as the standard deviation of the out-of-plane velocity fluctuations (`outOfPlaneStdDeviation`) (see table 2.3). Some particles may thus leave or enter the laser sheet, thus resulting in a total number of illuminated particles in the second image different from that in the first image. Note that the expected value of the difference in the initial and final  $N_i$  is zero since the out-of-plane motion is Gaussian with zero mean.

Each image spot, i.e., the imaged particle, has its size defined in pixels(px) and the image spot intensities are modelled as a Gaussian distribution of gray levels, essentially following the procedure of [17]. The user may choose the diameter of the image spot, from its fractional radius size ( $2.0 \times \text{particleRadius}$ ). The actual aspect of the spot depends on the position of its centre in the  $(x, y)$  plane and its lateral position - the light reflected by each particle depends on the incident light which, in turn, depends on its lateral position. A particle thus appears smaller if it is not exactly in the centre of the laser sheet. Note also that particles that are beyond the  $I_0/e^2$  still reflect light, but are so dim that its contribution is considered noise.

Each image is lossless TIF with 8 or 16 bit depth gray scale images, where only 8, 10 or 12 bits are used to encode the tracer image simulating the light intensity reflected by particles of specified diameters (see `bitDepths` description in section 2.4).

White Gaussian Image Noise (WGIN) is added to both images independently (`noiseLevel`). The synthetic images thus generated are very similar to true planar laser-illuminated tracer images obtained in laboratory environments, as seen in Figure 2.3a, Figure 2.5a and Figure 2.5b. Note however, that all image features are decided by the user and correspond to a zero-uncertainty ground truth. This configures an absolute novelty in what concerns synthetic image generators as the performance of evaluated algorithms can be directly assessed as a function of image parameters for which there is no uncertainty.

The application is supported by an internal library, so that the application parameters are translated into library parameters, while others are directly set by the application. Table 2.3 summarizes the parameters that can be set by the user in the application, as well as, the parameters that the library uses, but are not exposed by the application to the user. Should the end-user intend to create a custom application, then all library parameters are accessible, and they are documented in the technical manual.

Table 2.3: Configurable control parameters

| <i>Application input parameters</i>              |  |
|--|--|
| <code>displayFlowField</code>                    | Flag for optional generation of flow field preview |
| <code>closeFlowField</code>                      | Flag for automated closing of flow field figure    |
| <code>sizeX</code>                               | Target image horizontal resolution                 |
| <code>sizeY</code>                               | Target image vertical resolution                   |
| <code>bitDepths</code>                           | Image bit depth                                    |
| <code>flows</code>                               | List of flow type to generate                      |
| <code>deltaXFactor</code>                        | Displacement ratio relative to the IA size         |
| <code>particleRadius</code>                      | The particle radius in <i>px</i>                   |
| <code>Ni</code>                                  | Number particle concentration                      |
| <code>noiseLevel</code>                          | Noise Level in dBW                                 |
| <code>outOfPlaneStdDeviation</code>              | Out of plane standard deviation (mm)               |
| <code>numberOfRuns</code>                        | Number of runs (details in section 2.4)            |
| <i>Library constant input parameters</i>         |  |
| <code>flowParameters.maxVelocity</code>          | Maximum velocity in mm/s                           |
| <code>pivParameters.lastWindow</code>            | IA size for last window [px , px]                  |
| <code>pivParameters.laserSheetThickness</code>   | Laser sheet thickness (mm)                         |
| <code>imageProperties.mmPerPixel</code>          | Pixel to millimeters scale                         |
| <code>pivParameters.particleIntensityPeak</code> | Peak reflected light intensity                     |

## 2.4 Illustrative Examples

In this section, examples are provided for some parameters.

1. Figure 2.2a depicts an uniform flow field
2. Figure 2.2b refers to a Poiseuille flow field
3. Figure 2.2c represents the flow field for the Rankine vortex.
4. Figure 2.2d depicts the Rankine vortex with superimposed uniform flow field
5. Figure 2.2e shows an inviscid stagnation point flow field.

These are examples of, some of, the currently supported flow fields as implemented by the PIV image generator tool.

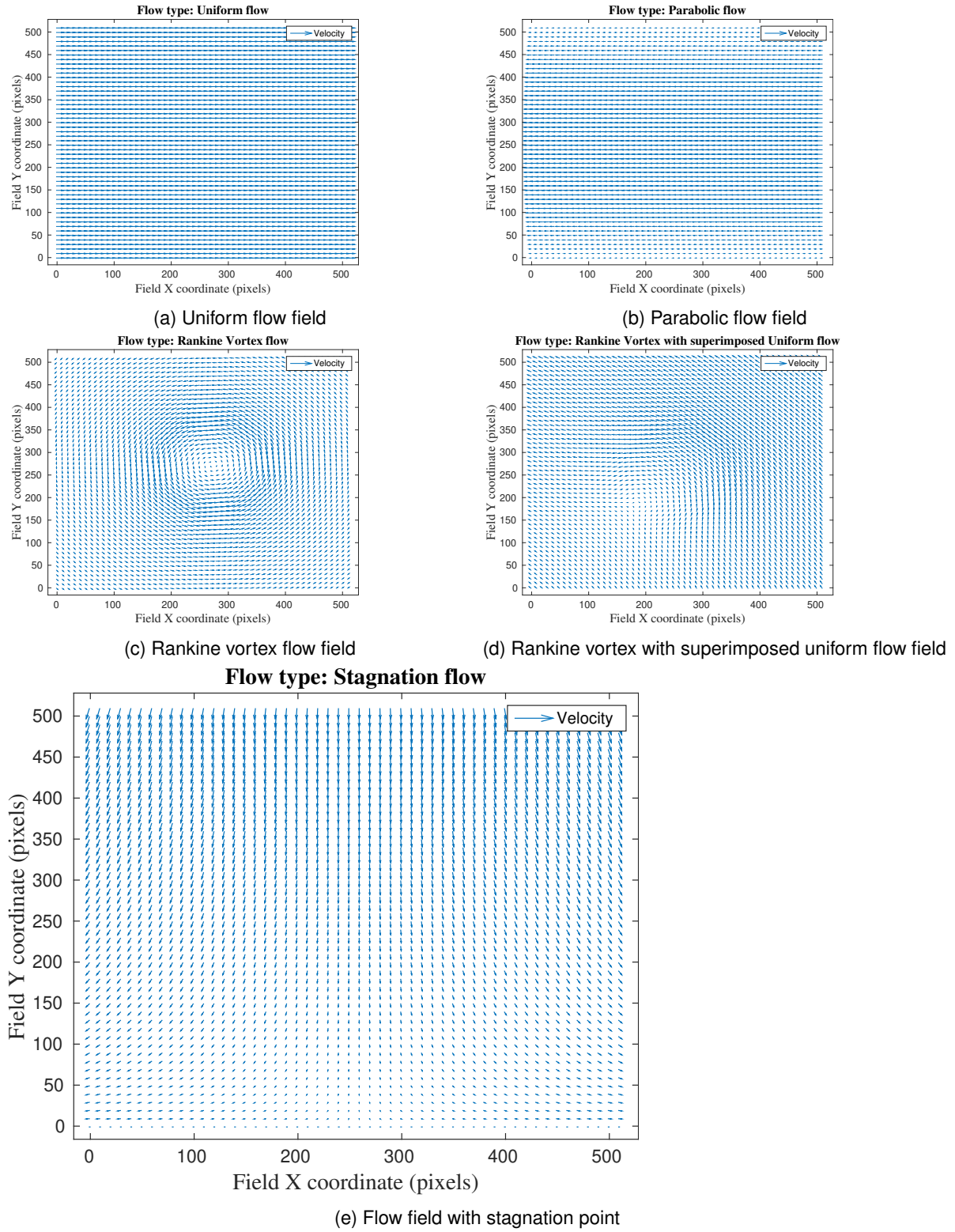
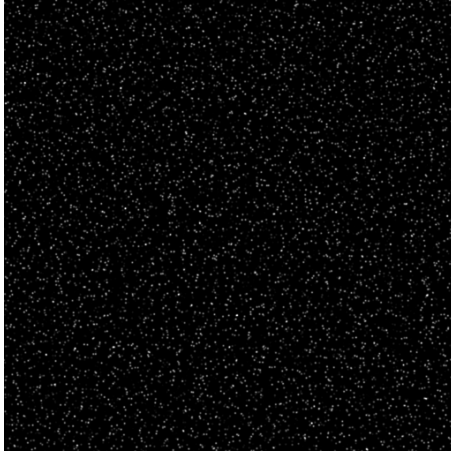


Figure 2.2: Flow types

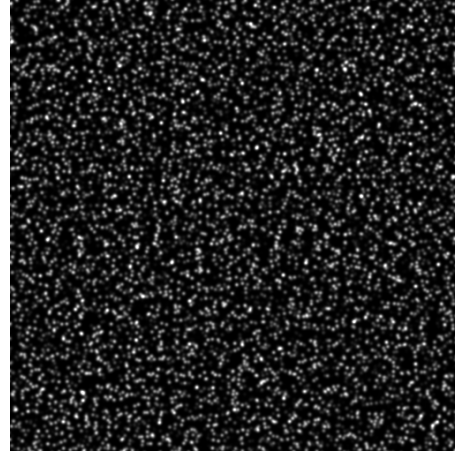
## Particle size

Figure 2.3a shows a sample synthetic PIV image with particle diameter of 3 px (pixels) and a number particle concentration by volume set to  $N_i = 6$ , while Figure 2.3b shows a sample synthetic PIV image

with particle diameter of 6 px and  $N_i = 6$ . Both images are shown as generated by the PIV image generator tool. Figure 2.5a depicts a synthetic PIV image with  $N_i = 12$ , while Figure 2.5b presents a synthetic PIV image with  $N_i = 16$ , both with particle diameters of 4 px and WGIN equal to 10 dBW.

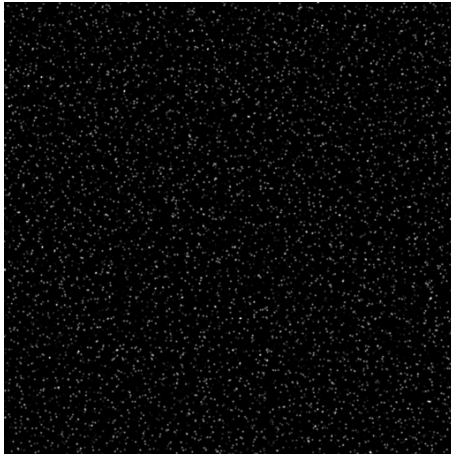


(a)  $N_i = 6$  and 3 px diameter particles

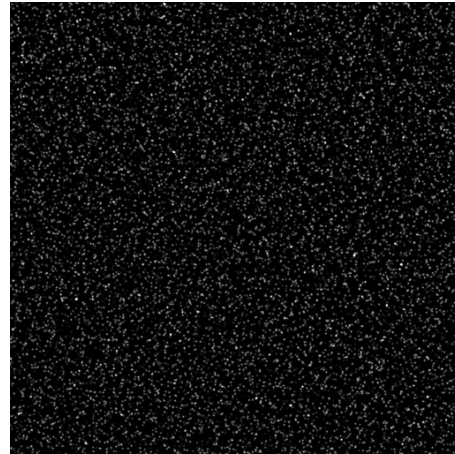


(b)  $N_i = 6$  and 6 px diameter particles

Figure 2.3: Different particles sizes, same concentration



(a)  $N_i = 6$  and 3 px diameter particles



(b)  $N_i = 12$  and 3 px diameter particles

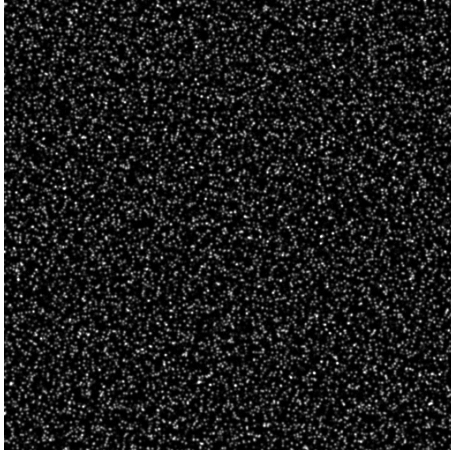
Figure 2.4: Same particles size, different concentrations

Figure 2.4a shows a sample synthetic PIV image with  $N_i = 6$ , while Figure 2.4b shows a sample synthetic PIV image with  $N_i = 12$ . Both images have particle diameter of 3 px (pixels) and are shown as generated by the PIV image generator tool.

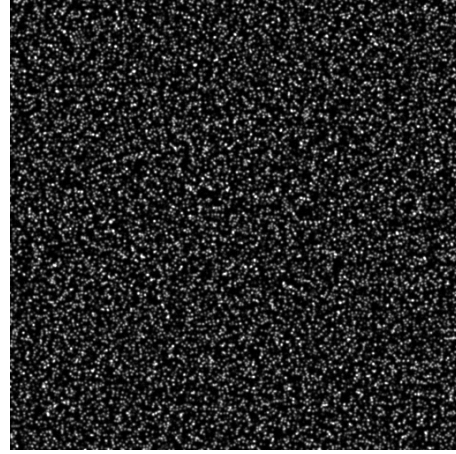
## User code sample

The following listing is a slightly modified, but functional version taken from `exampleAllTestImagesMain.m` in [14]:

```
sizeX=512; %Image width without margins
sizeY=512; %Image height without margins
flows={'uniform' 'parabolic' 'stagnation' 'shear' ...
```



(a)  $N_i = 12$ , WGIN 10 dBW and 4 px particles



(b)  $N_i = 16$ , 10 dBW WGIN and 4 px particles

Figure 2.5: Particle concentration and Noise effect

```
'shear_45d0' 'shear_22d3' 'rankine_vortex' ...
'decaying_vortex' 'rk_uniform'};
bitDepths=8;
deltaXFactor=[0.05 0.25];
particleRadius=[1.5 3.0];
Ni=[6 12];
noiseLevel=[0 15];
outOfPlaneStdDeviation=[0.025 0.10];
numberOfRuns=2;

generatePIVImagesWithAllParametersCombinations;
```

This configuration will generate images with 512 pixels  $\times$  512 pixels for each exhaustive combination of the parameters. This means that, e.g. Uniform flow will be generated for the image bit depth of 8 bits (because is the only value specified) and for each value of the displacement parameter `deltaXFactor`, combined with each value of the particle radius parameter `particleRadius` and so on. Upon exhausting the remaining combinations, the next flow is considered, until all flow options are exhausted.

1. `bitDepths` can be either 8, 10 or 12-bits and defines the pixel bit depth for the generated images. The pixel bit depth will affect the gray scale used to represent the brightness. Thus, the 8-bit grayscale will be in the range of 0 to 255 brightness values and the 8-bit TIF file format will be used to store the images. For 10-bit bit depths, the brightness values will range from 0 to 1023, and the 16-bit TIF format will be employed. The 12-bit bit-depth will have brightness values in the range of 0-4095, using 16-bit TIF images. For the 10-bit and 12-bit bit depths, where the 16-bit TIF are employed, it also implies that images will look darker, because the file format brightness/luminosity ranges from 0 (Black) to 65535 (White), but the generated images will have a much lower range, thus generic image viewer applications, that aren't aware of the intended use case, will display



dark images. If higher bit depths are needed, the code can be modified to support up to 16-bit, the current limit of TIF format.

2. `particleRadius` is the seeding particle radius in pixels.
3. The `deltaXFactor` defines in the maximum displacement as a proportion between the maximum displacement and the IA size. For instance, an IA of 16x16 with `deltaXFactor=0.25` will define a maximum velocity vector with magnitude 4.
4. The `noiseLevel` parameter specifies the WGIN (dB) values to consider. Like all the other parameters, can be a single value or an array of values of interest.
5. The `outOfPlaneStdDeviation` parameter defines the out of plane standard deviation and is related to the parameter `numberOfRuns`, in the sense that `numberOfRuns` should be greater or equal to the number of elements in `outOfPlaneStdDeviation`. For instance `numberOfRuns=2` will result in the generation of a single case of `outOfPlaneStdDeviation=0.025` and a single case of `outOfPlaneStdDeviation=0.10`. However if `numberOfRuns=7` then this would result in the generation of four images with `outOfPlaneStdDeviation=0.025` and three images with parameter `outOfPlaneStdDeviation=0.10`.

For the above listing, it generates  $9 \text{ flows} \times 1 \text{ bit-depth} \times 2 \text{ displacement factors} \times 2 \text{ particle radius} \times 2 \text{ particle concentrations (Ni)} \times 2 \text{ noise-levels} \times 2 \text{ out of plane values}$ , thus resulting in 288 total combinations, producing 288 synthetic PIV image pairs along with corresponding validation data.

## 2.5 Impact

The tool we have developed may be employed for a wide range of comparative studies of PIV and OF algorithms. It may help in standardizing source data sets for benchmarking the methods under test. Different research groups may use it to compare their results allowing direct cross-validation efforts.

The set of flow types can be easily extended to support additional flow types, by defining additional flow objects that abide to the defined methods signatures. Other possible extensions include importing displacements maps from simulation tools, like DNS-generated flow fields, from which images with different particle image parameters can be created. The image generator can also be adapted for other fields of applications as it can be controlled to create the appropriate data sets, rather than limiting to a single generic data set.

The tool has already been used in a comparative study of Optical Flow and PIV methods, [15, 16], where the image generator tool was used to generate a total of 7200 different synthetic PIV images, pertaining to 2160 different control parameter combinations, thus providing the test data for such comparative study. The most adequate particle sizes, in the sense that minimize velocity errors, and image properties were devised for each method. This example is representative but does not exhaust the possibilities of the tool.

A non-exhaustive list of challenges posed to PIV and Optical Flow algorithms are summarized next.

1. Identifying the correct deformation rates for different orientations of the shear flow, namely when the axes of the image are not aligned with the principal deformation axes nor with the referentials for which one velocity component is null. This might highlight problems in PIV algorithms associated with peak locking due to inadequate modelling of sub-pixel motion.
2. Reproducing the theoretical Poiseuille profile independently of the orientation of the flow. This test may be useful to detect peak locking bias and susceptibility to insufficient tracers in regions with strong gradients (near the wall).
3. Identifying the stagnation point in the axisymmetric stagnation point flow. This is a relevant test to assess the response of the algorithms as the tracer density is reduced.
4. Guarantying zero vorticity in the stagnation point flow. This may be relevant to assess the susceptibility of the algorithms to out-of-plane loss of pairs that may generate erroneous vectors leading to apparent vortical motion.
5. Identifying the rotational core in the Rankine vortex and in the Rankine vortex with superimposed uniform flow. In the former flow, streamlines are circular but the flow is rotational in the core region only, which cannot be envisaged by observing the streamlines. In the latter flow, which is similar to the observation of the vortex by an observer moving with uniform motion, the streamlines may not identify closed loops at all; however the Galilean invariance of the underlying vortex must be captured by the algorithms.
6. Identifying the centre of the decaying vortex. The velocity tends to zero in the centre of the vortex while the vorticity attains its peak value. This test may be challenging to assess the performance of the algorithms in low tracer densities.

A wide range of relevant performance tests can thus be designed with this tool. It is in this sense that we considered this set of flows sufficient for the purpose of testing planar PIV and Optical Flow algorithms. Furthermore, we argue that the tool is particularly adequate for validation in the sense that it provides a zero-uncertainty ground truth in the form of continuous velocity fields to compare with the results of PIV and OF algorithms.

## 2.6 Conclusions

We have presented an image generating tool aimed at creating images that simulate those obtained in a natural laboratory PIV environments, resulting in spots of illuminated tracers against a dark (but normally noisy) background. Image characteristics and error sources are fully parameterized. Images are generated from zero-uncertainty velocity and displacement fields. This configures the key novelty of the tool: fully parameterized image generation and zero-uncertainty ground truth is a combination of features never before conjoined in synthetic particle image generators.

Data is exported both for PIV and Optical Flow methods. The tool is extensible, since it is open-source, it can be modified by others, and can be easily modified to support additional flow fields. the values of the control parameters can be easily adjusted, using the provided sample scripts.

The tool was already used for a comparative study of Optical Flow and PIV methods applied to tracer based images for fluid mechanics velocimetry applications

We strongly believe that this tool can be used for additional studies for similar applications, even helping in standardizing the source data sets for benchmarking the methods under test. The image generator can also be adapted for other fields of application.

## 2.7 Credit

Luís Mendes: Methodology, Software, Validation, Writing - Original Draft

Alexandre Bernardino: Conceptualization, Validation, Writing - Review & Editing, Supervision

Rui M. L. Ferreira: Conceptualization, Validation, Writing - Review & Editing, Supervision

## 2.8 Conflict of Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

## Acknowledgements

This work is supported by the PhD grant SFRH/BD/137967/2018 from the Portuguese Foundation for Science and Technology (FCT).

## 2.9 References

- [1] Particle-Imaging Techniques for Experimental Fluid Mechanics. Annual Review of Fluid Mechanics 23, p.261-304 (1992). <https://doi.org/10.1146/annurev.fl.23.010191.001401>
- [2] Adrian, R. Twenty years of particle image velocimetry. Springer Verlag (2005). <https://doi.org/10.1007/s00348-005-0991-7>
- [3] Batchelor, G.K. An Introduction to Fluid Dynamics. Cambridge University Press p.286 (2000).
- [4] Berthe, A., Kondermann, D., Garbe, C., Affeld, K., Jähne, B, Kertzscher, U. The Wall-PIV Measurement Technique for Near Wall Flow Fields in Biofluid Mechanics. Imaging Measurement Methods for Flow Analysis p.11-20 Springer Berlin Heidelberg (2009). doi: 10.1007/978-3-642-01106-1\_2

- [5] Blocken, B., Druenen, T., Toparlar, Y., et. al. Aerodynamic drag in cycling pelotons: new insights by CFD simulation and wind tunnel testing. *Journal of Wind Engineering and Industrial Aerodynamics* 179 p.319-337 (2018). <https://doi.org/10.1016/j.jweia.2018.06.011>
- [6] Carlier, J., Wieneke, B., Report 1 on production and diffusion of fluid mechanics images and data. Tech. rep., Fluid image analysis and description (FLUID) Project (2005). URL <http://fluid.irisa.fr/data-eng.htm>
- [7] Chao, H., Gu, Y., Napolitano, M. Survey of Optical Flow Techniques for Robotics Navigation Applications. *Journal of Intelligent and Robotic Systems* 73 p.361-372 (2014). <https://doi.org/10.1007/s10846-013-9923-6>
- [8] Garcia, F., Cerri, P., Broggi, A., Escalera, A., Armingol, J. Data fusion for overtaking vehicle detection based on radar and optical flow. *2012 IEEE Intelligent Vehicles Symposium* p.494-499 (2012). doi: 10.1109/IVS.2012.6232199.
- [9] Hongwei, W., Zhan, H., Jian, G., Hongliang, X. The Optical Flow Method Research of Particle Image Velocimetry, *Procedia Engineering* 99 p.918-924 (2015). <https://doi.org/10.1016/j.proeng.2014.12.622>
- [10] Gerontakos, P., Lee, T. PIV study of flow around unsteady airfoil with dynamic trailing-edge flap deflection. *Exp Fluids* 45(955) (2008). <https://doi.org/10.1007/s00348-008-0514-4>
- [11] Liu, T., Merat, A., Makhmalbaf, M.H.M., Fajardo, C., Merati, P. Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. *Exp Fluids* 56(166) (2015).
- [12] Ma, C.M., Wang, J.X., Li, Q.S., Qin, H. Vortex-Induced Vibration Performance and Suppression Mechanism for a Long Suspension Bridge with Wide Twin-Box Girder. *Journal of Structural Engineering* 144(11) (2018). [https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0002198](https://doi.org/10.1061/(ASCE)ST.1943-541X.0002198)
- [13] Mase, K. Recognition of Facial Expression from Optical Flow. *IEICE transactions on Informations and Systems*. E74-D(10) p.3474-3483 (1991).
- [14] Mendes, L., Ferreira, R.M.L., Bernardino, A. Particle Imaging Velocimetry (PIV) image generator [Source Code]. <https://doi.org/10.24433/CO.1931173.v3> (2020).
- [15] Mendes, L., Ricardo, A, Bernardino, A., Ferreira, R.M.L. PIV without cross-correlation: an assessment of optical flow methods. *6th IAHR Europe Congress* (2020).
- [16] Mendes, L., Ricardo, A, Bernardino, A., Ferreira, R.M.L. Comparison of PIV and Optical flow for river flow applications. *River Flow 2020* (2020).
- [17] Raffell, M., Willert, C., Scarano, F., Kähler, C., Wereley, S., Kompenhans, J. Particle Image Velocimetry - A practical guide. *Springer* (2018). <https://doi.org/10.1007/978-3-319-68852-7>

- [18] Ricardo, A.M., Koll, K., Franca, M.J., Schleiss, A.J., Ferreira, M.L. The terms of turbulent kinetic energy budget within random arrays of emergent cylinders. *J. Water Resources Research* 50(5) p.4131-4148 (2014). <https://doi.org/10.1002/2013WR014596>
- [19] Schmidt, B.E, Sutton, J.A. High-resolution velocimetry from tracer particle fields using a wavelet-based optical flow method, *Exp Fluids* 60(37) (2019). doi: 10.1007/s00348-019-2685-6
- [20] Soid, S.N., Zainal, Z.A. Spray and combustion characterization for internal combustion engines using optical measuring techniques – A review. *J. Energy* 36(2) (2010). <https://doi.org/10.1016/j.energy.2010.11.022>



## Chapter 3

# A comparative study of optical flow methods for fluid mechanics

Article published in the Experiments in Fluids journal with DOI: <https://doi.org/10.1007/s00348-021-03357-7>

Received 04 June 2021, Revised 30 September 2021, Accepted 15 November 2021, Published 15 December 2021.

### 3.1 Abstract

We present a benchmark study of Optical Flow (OpF) methods for fluid mechanics applications. It is aimed at assessing the performance of three OpF methods, Lucas and Kanade (in: Proceedings of the 7th 1519 international joint conference on artificial intelligence, 1981), Horn and Schunck (AI 17:185–203, 1981) and Farnebäck (Two-frame motion estimation based on polynomial expansion, in: Bigun, Gustavsson (eds) Image analysis, Springer, 2003), combined or not with the Liu and Shen (JFM 614:253–291, 2008) algorithm. The performance of the OpF methods, evaluated exclusively as the difference between the values of the methods and of ground-truth (reference values), is benchmarked for deformation dominated, rotation-dominated and uniform flows. For each flow type, relative and absolute errors are computed for different tracer displacements, noise levels, pixel particle sizes, image bit-depths and particle concentrations. The accuracy of the OpF methods seems mainly affected by the magnitude of velocity gradients and convective accelerations. It does not seem to be affected by the relative preponderance of rotational and deformation components. The inner region of the Poiseuille flow and the saddle point in the Rankine vortex combined with uniform flow pose significant difficulties to all methods. The performance of the Lucas-Kanade/Liu-Shen combination is the best for all flow types, image conditions and image bit depths. The Farnebäck/ Liu-Shen combination has a similar high performance but only for image depths of 10 bit or higher. These OpF methods maintain a high performance for tracer sizes and concentrations outside the PIV optimal range. Horn-Schunck is the worst performing method, due to high sensitiveness to particle concentration variations or particle sizes. These results can be

used to plan new particle-based velocimetry experiments or to retrieve further information from existing PIV databases.

Keywords: Optical flow Particle Image Velocimetry Benchmark

## 3.2 Introduction

Optical Flow (OpF) may be defined as the two-dimensional displacement field that describes the apparent motion of three-dimensional motives projected in the imaging plane – [3]. It is normally determined by evaluating changes in the position of brightness patterns between two successive images.

Several OpF methods have been developed for machine vision applications, based on different concepts and employing different strategies. The most common include differential techniques, e.g. [4], [5], [6], [7], region-based and feature-based matching [8]; [9]; [10], global matching [11]; [12] spatio-temporal filtering [13], [14] and, more recently, convolutional neural networks [15]; [16]. Other machine learning (ML) techniques include statistical learning as in [17]; [18]; [19]. Methods are often termed sparse or dense. The former if flow vectors are given only for some image features such as corners or edges, e.g. [8], here designated as sparse L-K; the latter (dense) if flow vectors are found for all image pixels, e.g. [5] (here designated as dense version of L-K or L-K) and [7]

The performance of OpF methods for machine vision, mostly in terms of accuracy and precision, has been assessed by different methods, generally involving synthetic or real images, for which the underlying camera movements are simple (e.g. zoom, pan or dolly) and known a priori, and specific metrics [3, 20–22]. Evaluation metrics comprise absolute and relative errors, including linear and angular errors and other metrics specific for machine vision. There is still considerable debate on the most appropriate metrics for different applications [22] but it is relatively consensual that OpF algorithms must, at some point in their development, be tested against synthetic results for which uncertainty is very low [23] – the ‘ground truth’ – or statistically well characterized.

In the scope of fluid mechanics and, more specifically, in the context of tracer-based image velocimetry, OpF has been described as Particle Image Velocimetry without correlation [24, 38]. It was first employed by [17], who proposed an Orthogonal Dynamic Programming (ODP) method, a global matching technique that determines optimal alignments between imaged patterns. [17] tested the method with synthetic and real PIV images and reported improvements in the accuracy of flow field (quantified by a reduction of absolute and angular errors), opening the possibility of increasing the spatial resolution of the velocity field. [18] adopted a simpler differential technique featuring a variational approach to globally minimize the magnitudes of spatial gradients of both velocity components. The method is simpler than that of [17] and was shown to perform better for noisy images.

Fluid flows are theoretically described by physical laws. If understood as continua, their observable kinematic features are a result of their dynamics, governed by the continuity and Navier-Stokes equations. It is thus reasonable to introduce physically-based constraints, particularly for differential approaches. Along this line of thought, the formal relation between optical flow and the motion of a continuum fluid, seeded with tracers, and illuminated by coherent light has been investigated by [25]. Their



optical flow equation is essentially a two-dimensional conservation equation of brightness featuring a source term that includes light scattering and absorption, the consequences of optical arrangement and light sheet thickness and the effect of particle diffusion. A fundamental result is that the optical flow is proportional to the weighted average of the path-averaged velocity of tracers in the illuminated volume (the latter is corrected by the concentration of tracers). The [25] method is often called 'physics-based' optical flow, a shorthand we also employ in this text.

The advances of [25] resulted in equations of motion projected in the image plane that have been embedded in some optical flow algorithms [29, 30]. Other approaches, more closely following the link between observed motion and fluid dynamics, have been proposed, for instance to resolve the ill-posed nature of the differential approach. In particular, appropriate formulations of the continuity equation have been tested as alternatives to the brightness constancy assumption and smoothness constraint [19]. [20] considered Stokes flow approximation (a particular solution of the Navier-Stokes equations) as a priori knowledge. The theoretical applicability of these approaches is limited: the formulation of [19] is exactly valid when light radiance is proportional to an integral of the fluid density across a measurement domain, which is not always true, in particular in incompressible fluids; and the Stokes flow approximation is valid as a first order approximation in turbulent flows. In order to overcome these limitations, [21] and [22] argued for the inclusion of a predictor term, acting as a spatio-temporal smoothing filter, based on a simplified equation of conservation of vorticity, derived from the integration of the Navier-Stokes over the thickness of the laser sheet, in planar PIV set-ups. [35], followed a similar approach but introduced an extra step, a backward integration of an adjoint evolution model. Results were seen to improve, [22], but at the expense of computational power. Other approaches included [36], who resolved the ambiguity inherent to the Horn-Schunck constant brightness approach with a sub-grid model inspired by the LES (Large Eddy Simulation) decomposition, or [37], that included small turbulent scales in the optical flow constraint equation through a stochastic, but physically-based, transport equation.

Some OpF approaches explicitly avoid multi-resolution based on a "pyramidal" image structure, e.g. [38], [39]. Instead, they decompose the flow into scales, so that different optimization steps are carried out for different energy content or other scale metrics, e.g. [21], [35], as is the case for wavelet decomposition [40], [2]. However, "pyramidal" multi-resolution have been advocated by several researchers, e.g. [41], [42], [18], [43], [24]. [37] propose that some of the shortcomings of the pyramidal multi-resolution approach can be eliminated by Gaussian filtering the image pair, before warping, at every level of the pyramidal image sub-sampling structure. [45] showed that the low-pass filtering and several pyramid levels are needed for complex multi-phase flows.

Multi-resolution, involving successive filtering and warping steps, particularly when dense OpF estimates are needed [23], may result in expensive computations. Implementation thus becomes a relevant issue. The dense version of the Lucas-Kanade algorithm has been seen to be appropriate for massively parallel implementations, making use of Graphics Processing Units (GPUs) [24, 47] or field-programmable gate arrays (FPGAs) [48]. For real-time applications, the eFolki algorithm [49] is based on a massively parallel Lucas Kanade implementation (for multi-GPU), featuring pyramidal multi-resolution, image warping, reduction of iterative steps but also a local polynomial expansion of image brightness

that contributes for smoother velocity gradients. Other algorithms have been subjected to massively parallel optimization, including [7] but they have not yet been implemented in the scope of fluid mechanics applications [50], [7].

For correlation-based PIV, the optimal imaged tracer displacement between two consecutive images is about 25 % of the size of the interrogation window, (e.g. Westerweel 24). Such displacement may be too large for OpF methods. If, additionally, imaged tracers are too small or too few, OpF may return erroneous or noisy velocity fields. This problem may not be corrected by successive low-pass filtering and sub-sampling associated to pyramidal multi-resolution and may benefit from an initial cross-correlation step [30]. Hybridization of correlation-based PIV and OpF has been tested as a natural way to increase the resolution of planar digital PIV (achieving, in the limit, a velocity vector per pixel). [28] suggested refining the correlation-based PIV without sub-pixel interpolation using the [25] estimator. They reported the improvement of accuracy with respect to correlation-based PIV, and highlighted the benefits of higher resolution near the boundaries. [29] envisaged methods to increase resolution of not only traditional PIV but also Particle Tracking Velocimetry (PTV). The hybridization proposal of [53] addressed specific problems for OpF arising from classical PIV solutions. In particular [53] tackled the problem of variations in laser intensity (common in commercial pulsed-laser PIV hardware) by using the assumption of constant brightness gradient rather than brightness intensity, in a differential algorithm akin to Horn-Schunck's. The issue of varying brightness was also addressed by [32] and [33]. They proposed an algorithmic implementation of a hybrid method featuring a correlation-based PIV and [25] OpF method, modified to deal with large image tracer displacements.

Combination of different OpF methods has also been proposed, as a means to save computational time in the case of dense estimators. For instance, [30] use the Horn-Schunck estimator for an initial solution and the [25] estimator for a refined solution.

All OpF methods for fluid mechanics applications have been tested with synthetic or real PIV images. Well-known data sets include the 'standard' images produced by the Visual Society of Japan [56], based on a Large Eddy Simulation (LES) database of an impinging jet flow, which have been employed by [18] or [28], among others, or the synthetic and real datasets from the "International PIV-Challenge", [57–59]. Some of the synthetic images of the latter were generated with the EuroPIV synthetic image generator [60]. This image generator has also been used to produce synthetic PIV images from idealized flows [24] or Direct Numerical Simulation (DNS) results, e.g. [21], [2]. Other standard images and DNS datasets have also been proposed by [1]. Alternative synthetic images of idealized flows have been used, among others, by [30], [33] (Oseen vortex pair), [53] (Rankine vortex). Simpler, low-Reynolds (low-Re), two-dimensional Navier-Stokes solutions, featuring high vorticity flows, have also been converted into synthetic PIV images by [40] or [37]. In the cases above, there is no uncertainty in the position of the tracers pictured in each image pair - the velocity and displacement fields can be considered ground truth. To address the challenges associated to real PIV images, it is also common to use real datasets to verify OpF algorithms, e.g. [36], [40], [49], [53], [33]. Images taken from machine vision benchmarks have also been used to validate algorithms mostly used in the context of fluid mechanics, e.g. [17], [49].

To summarize, OpF is an established technique to determine the apparent motion of image se-

quences, for which there are several methods benchmarked and compared for machine vision applications. It has been tentatively applied for particle-based velocimetry for more than two decades. Currently, some algorithms and implementations have been validated with synthetic and laboratory datasets for fluid mechanics applications. It has been shown that OpF can be hybridised with planar PIV, thus overcoming limitations of OpF and potentially increasing PIV resolution. OpF methods have been tested individually but not compared in a common framework. To the best of our knowledge, there are no studies assessing the performance of different OpF algorithms for different types of flow and for images of different quality. We believe this has hindered OpF to assume a central role in fluid mechanics.

To promote the growth of the use of OpF in fluid mechanics, we propose that the adequateness of different methods for particle-based velocimetry must be further investigated in a common framework. Considering that the rate of change of any general flow field can be decomposed into a rate of rotation and a rate of deformation, we observe that it is not known whether some OpF algorithms are better suited for some types of flow (rotational or deformation-dominated) and why, if that is the case. It is also not known if some OpF approaches are more resilient than others to image noise or insufficient tracer number density or quality and if this resilience changes with the type of flow.

Addressing these issues, the aim of this paper is to conduct a benchmark study of the OpF methods most frequently used for fluid mechanics applications – [4], [5], [25] – to which we add the [7] method. The latter is tested for the first time in the specific context of tracer-based velocimetry.

The benchmark study considers a parametric space whose main axes are: i. the type of flow; ii. imaging conditions and iii. image characteristics. The comparative exercise involves all combinations of OpF methods and a reference implementation of correlation-based PIV. All test cases are based on synthetic images expressing particle-laden flows that are realistic but for which there are theoretical solutions. Thus, we generated an ideal scenario of zero uncertainty (denoted ‘ground-truth’) which is used to compute the absolute and relative accuracy of OpF and PIV methods. In this benchmarking exercise, performance is evaluated exclusively in terms of accuracy – the difference between the values of the OpF methods and of ground-truth.

In what concerns the types of flow, we considered: a) deformation-dominated flows such as the Poiseuille’s and a irrotational flow with a stagnation point, b) rotation-dominated flows such as the Rankine vortex with and without a superimposed uniform flow, and c) a simple uniform flow as it is present in other studies, such as [3].

We took into account that PIV databases have diversified characteristics, namely tracer spot size, number density of imaged particles, mean brightness and ambient noise, size of the field of view, camera resolution, thickness and intensity of the laser beam, focus of the laser sheet, among other factors. We generated different PIV data-sets parameterized by the parameters above, except optical and laser conditions (which we fix) and out-of-plane motion (which we treat stochastically).

For each OpF method (and for the PIV), we generated approximately 6400 data-sets, corresponding to meaningful combinations of types of flow and different values of the control variables expressing imaging conditions and image characteristics. All data-sets were generated by the synthetic image and flow generator described in [61]. We have discarded the use of Direct Numerical Simulation (DNS)

databases to generate reference values. The advantage of a realistic velocity field is not significant in this case given that, in this work, we are mostly concerned with quantifying errors across a wide parametric space involving combinations of different types of flow and image characteristics.

The main methodological features and the key findings of the paper are described in five sections, after this introduction. The synthetic image and flow generator are described in section 2, along with the identification of the flows considered for the benchmark. Section 3 describes the data-sets and the evaluation metrics. Section 4 is dedicated to the comparison of the performance of the different OpF methods across the parametric space. Section 5 offers a discussion of the main findings.

### 3.3 Synthetic image and flow generator

#### 3.3.1 Flow field generation

The performance of all OpF methods is assessed by quantifying the errors (relative and absolute) between the velocities determined by the methods and the true velocities whose generation process is described herein. We generate both uniform flows and non-trivial flows, with different combinations of rotation and deformation: Poiseuille flow (see Figure 3.1 a), inviscid stagnation point flow (see Figure 3.1 b), Rankine vortex (see Figure 3.1 c) and Rankine vortex with superimposed uniform flow (see Figure 3.1 d).

These flows pose specific challenges to the velocimetry algorithms, including (i) large displacements in all points (Uniform flow) or in parts of the image (all other flows), (ii) strong wall gradients (Poiseuille flow), (iii) singularities such as a stagnation point (inviscid flow with stagnation point) and the core of rotational flows (Rankine vortex), (iv) regions of high convective accelerations (inviscid flow with stagnation point, Rankine vortex with superimposed uniform velocity field) and (v) saddle points (of Rankine vortex with superimposed uniform flow).

The displacement and velocity fields of all flows have been generated as described in [61]. The displacement fields are, in all but one case, analytic solutions of the set of differential equations that describe the velocity field. The only exception is the Rankine vortex with superimposed uniform flow for which a 4<sup>th</sup> order Runge-Kutta scheme was used to solve the set of ODEs. The equations expressing flow kinematics are presented in Table 3.1.

For PIV, a single ground-truth velocity vector is exported at the center of each interrogation area. For OpF methods -all dense-, velocity vectors are exported at the center of each pixel.

#### 3.3.2 Synthetic image generation

The PIV synthetic images were generated as described in [61]. As control variables, we consider the imaging conditions and image characteristics identified in Table 3.2.

Table 3.1: Flow types, velocity flow fields and displacement equations. The displacements in the flow field are analytically solved from the corresponding velocity flow field expression. An exception is made for the Rankine vortex with superimposed uniform flow, for which the solution is obtained numerically by a 4<sup>th</sup> order Runge-Kutta scheme. In these equations,  $x$  and  $y$  are Cartesian orthogonal spatial coordinates specifying the position of a particle in a plane,  $x_0$  and  $y_0$  are the initial particle positions,  $r$  is the radial polar coordinate,  $\theta$  is the azimuthal polar coordinate,  $\theta_0$  is an initial azimuthal particle position,  $t$  is the time elapsed since an initial instant ( $t = 0$ ),  $x_c$  and  $y_c$  are the coordinates of the stagnation point, for the inviscid stagnation point flow, or the centre of the vortex, for Rankine vortices,  $\Gamma$  is the circulation of rotational core in the Rankine vortices,  $y_{max}$  and  $y_{min}$  are the coordinates of the upper and lower bounds of the flux tube within which the Poiseuille flow takes place,  $R$  is the radius for which maximum azimuthal velocity is attained in the Rankine vortex flows,  $m$  is a reference velocity,  $M$  is a scaling parameter that allows for the specification of the desired maximum relative displacement (a percentage of the IA size, always less than or equal to 25%). For the Rankine Vortex and Rankine with superimposed Uniform flow cases  $\Gamma = m2\pi R$ .

| Flow type                  | Displacements in the flow field   | Velocity flow field   |
|----------------------------|---|---|
| Uniform                    | $\begin{cases} x_1 = \frac{m}{M}t + x_0 \\ y_1 = y_0 \end{cases}$   | $\begin{cases} u = \frac{m}{M} \\ v = 0 \end{cases}$  |
| Poiseuille                 | $\begin{cases} x_1 = \frac{m}{M}(y_0 - y_{min})(y_0 - y_{max})t + x_0 \\ y_1 = y_0 \end{cases}$   | $\begin{cases} u = \frac{m}{M}(y - y_{min})(y - y_{max}) \\ v = 0 \end{cases}$  |
| Stagnation                 | $\begin{cases} x_1 = e^{\frac{m}{M_x}t}(x_0 - x_c) + x_c \\ y_1 = e^{-\frac{m}{M_y}t}(y_0 - y_c) + y_c \end{cases}$   | $\begin{cases} u = \frac{m}{M_x}(x - x_c) \\ v = -\frac{m}{M_y}(y - y_c) \end{cases}$   |
| Rankine vortex             | $\begin{cases} x_1 = r \cos(\frac{\Gamma}{2\pi R} \frac{1}{MR}t + \theta_0) \\ y_1 = r \sin(\frac{\Gamma}{2\pi R} \frac{1}{MR}t + \theta_0) \end{cases}, r \leq R$<br>$\begin{cases} x_1 = r \cos(\frac{\Gamma}{2\pi R} \frac{R}{Mr^2}t + \theta_0) \\ y_1 = r \sin(\frac{\Gamma}{2\pi R} \frac{R}{Mr^2}t + \theta_0) \end{cases}, r > R$ | $u_\theta(r) = \begin{cases} \frac{\Gamma}{2\pi R} \frac{r}{MR} & , r \leq R \\ \frac{\Gamma}{2\pi R} \frac{R}{Mr} & , r > R \end{cases}$   |
| Rankine vortex and Uniform | Numerically solved by 4 <sup>th</sup> order Runge-Kutta   | $\begin{cases} u = -\frac{\Gamma}{2\pi R} \frac{y}{MR} + u_0 \\ v = \frac{\Gamma}{2\pi R} \frac{x}{MR} + v_0 \end{cases}, r \leq R$<br>$\begin{cases} u = -\frac{\Gamma}{2\pi R} R \frac{y}{M(x^2+y^2)} + u_0 \\ v = \frac{\Gamma}{2\pi R} R \frac{x}{M(x^2+y^2)} + v_0 \end{cases}, r > R$ |

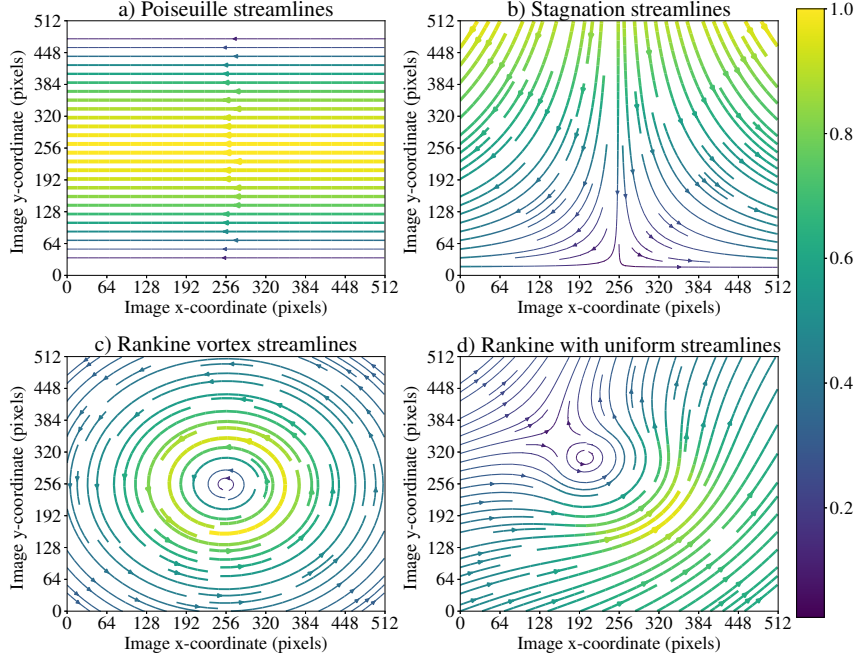


Figure 3.1: Streamlines for the Poiseuille and inviscid stagnation point flows. Flow fields generated with the corresponding equations shown in Table 3.1. Thicker lines indicate higher velocities. Color code varies from dark blue (low velocity) to yellow (high velocities). All velocities are normalized by the highest velocity in each flow type.

Table 3.2: Configurable image control parameters. Here,  $d_\tau$  is the spot diameter of the imaged tracer particle,  $N_i$  is the number of initial in-plane particles in each interrogation area (particles whose centres are within the virtual laser sheet thickness),  $\Delta D$  is the maximum in-plane displacement,  $b$  is the image bit-depth,  $N$  is the power of the White Gaussian Image noise (WGIN),  $\sigma_Z$  represents the standard deviation of the out-of-plane displacement (whose mean is zero),  $A^2$  is the image resolution,  $\Delta Z_0$  is the laser sheet thickness,  $I_0$  is the particle reflection peak intensity,  $f_u$  is the image scale factor, and  $IA$  is the size of the interrogation area for PIV measurements. The laser sheet thickness,  $\Delta Z_0$ , is defined as the normal distance between two parallel planes, on either side of the central plane, for which the light intensity is  $1/e^2$  of the peak (central) laser light intensity.

| <i>Variable parameters</i>            |  |
|---------------------------------------|--|
| Flow types                            | uniform, parabolic, stagnation, Rankine vortex and Rankine+uniform   |
| $d_\tau$                              | 1.0 px, 2.0 px, 3.0 px and 6.0 px  |
| $N_i$                                 | 1, 6, 12 and 16 particles  |
| $\Delta D$                            | 0.8 px, 1.6 px and 4.0 px  |
| $b$                                   | 8-bit, 10-bit and 12-bit   |
| $N$                                   | 0 dBW ( $\mathcal{N}(\mu = 0.0, \sigma^2 = 1.0)$ ), 5 dBW ( $\sigma^2 = 3.16$ ) and 15 dBW ( $\sigma^2 = 31.62$ ) for 8-bit depth (0 – 255 pixel intensity levels) and normalized for 10-bit, or 12-bit depths |
| $\sigma_Z$                            | 0.025 mm, 0.050 mm and 0.100 mm  |
| <i>Fixed parameters(in this work)</i> |  |
| $A^2$                                 | 512 px $\times$ 512 px   |
| $\Delta Z_0$                          | 2.0 mm   |
| $n$                                   | 10   |
| $I_0$                                 | 150.0  |
| $f_u$                                 | $7.5 \times 10^{-2}$ mm/px   |
| $IA$                                  | 16 px $\times$ 16 px   |

The fixed parameters in Table 3.2 are used to model the size of the field of view, the camera resolution, focal length and characteristics of the laser sheet, namely its thickness, its spatial (Gaussian) distribution, and its power (resulting in the peak image brightness).

The generation of an image pair is initialized by randomly placing tracer particles in each interrogation area, with a uniform distribution both for the in-plane and out-of-plane initial positions, such that the initial  $N_i$  value is respected. Then, the in-plane movement is computed, for all particles, from the flow field displacement equations in Table 3.1. The maximum displacement in the image, in any of the planar directions, is  $\Delta D$ . This maximum displacement restriction is enforced by the flow displacement equations in the form of the  $M$  parameter in Table 3.1, which must be set in a compatible manner to the  $\Delta D$  value. Then, the out-of-plane movement is calculated stochastically. The value of the standard deviation,  $\sigma_Z$  (see Table 3.2) is used to randomly generate a number, following the Gaussian distribution  $\mathcal{N}(\mu = 0, \sigma^2 = \sigma_Z^2)$ . This number is then added to the initial particle  $z$ -position.

Finally, the images are rendered according to the specified bit depth, after which, WGIN is added to the images. The light intensity scattered by the particles that reach the virtual camera sensor is modelled by a virtual light sheet, which in our case, has a Gaussian distribution with 2.0 mm thickness. The particles in the virtual laser light sheet scatter light at peak intensity at the center and appear gradually less bright as they move away from the center ( $z$ -component wise). Examples of the rendered images can be seen in Figure 3.2.

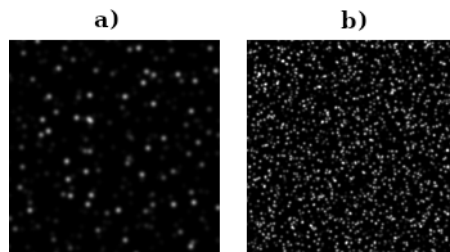


Figure 3.2: PIV image samples (164 px x 164 px) generated with the PIV Image Generator [61]. a) Imaged particle spot sizes of 6.0 px diameter and  $N_i$  value of 1, b) imaged particle spot sizes of 3.0 px diameter and  $N_i$  value of 16

To attain statistical relevance, each entry of the data-set is instantiated  $n$  times. Each entry of the data-set is called a Statistical Image Group (SIG) and corresponds to specific values of the controlling parameters in Table 3.2 repeated  $n$  times with varying  $\sigma_Z$ . In this work, for  $n = 10$ , there will be four images with  $\sigma_Z = 0.025$  mm, three images with  $\sigma_Z = 0.050$  mm and three images with  $\sigma_Z = 0.100$  mm. The varying  $\sigma_Z$  values in a SIG were introduced to reduce the combinatorial explosion of image pairs while retaining the key feature of noise generated by out-of-plane motion.

For more details about the image generation process and data export, please see [61]. The latest image generator tool can be downloaded from <https://git.qoto.org/CoreRasurae/piv-image-generator>.

## 3.4 Benchmark – data, algorithms and metrics

### 3.4.1 Benchmark data

The possible values of the parameters that characterize the synthetic images are summarized in Table 3.2. A total of 6480 unique parameterizations ( 5 flow types  $\times$  4 imaged particle spot sizes  $\times$  4  $N_i$  values  $\times$  3 displacements  $\times$  3 WGIN values  $\times$  3  $\sigma_Z$  values  $\times$  3 bit-depths), resulting in 2160 individual SIGs that are generated from all the possible combinations of parameters sections. This results in a total of 21600 image pairs (5 flow types  $\times$  4 imaged particle spot sizes  $\times$  4  $N_i$  values  $\times$  3 displacements  $\times$  3 WGIN values  $\times$  10 image pairs in a SIG  $\times$  3 bit-depths).

### 3.4.2 Methods and their implementation

The OpF algorithms used in this work were prepared by the first author as Python scripts and OpenCL code, [62] MATLAB reference implementation of Liu-Shen algorithm. It should be noted that our implementations were not created with the purpose of achieving the best possible performance, but to be independent from third party libraries that can change between releases, thus potentially impacting the results. They form an open-source reference library on their own, although with multiple different licences. The resulting scripts can be downloaded from <https://github.com/CoreRasurae/OpticalFlow-RI>.

Our custom PIV reference implementation was benchmarked against other PIV implementations [63]. The PIV application was developed in Java and OpenCL through Aparapi framework (<https://aparapi.com/>), namely Aparapi v3.0.0 [64] and is a work in progress towards a full featured PIV application, however it is not publicly available at this moment.

We benchmark the following algorithms: Lucas-Kanade [5]; Farnebäck [7]; Horn-Schunck[4]. These were combined with Liu-Shen [25]. All methods and combinations of methods were implemented in their simple versions and with a 2-level multipass pyramidal version

Our implementation of PIV consisted of a Grid Refining Scheme, as described in [38], pp.162-163 with IA sizes starting at 128 px  $\times$  128 px and ending with 16 px  $\times$  16 px, in 4 adaptive steps without superposition. For sub-pixel interpolation, a 3-point 1D-1D polynomial Gaussian was applied to the correlation output, centered at the correlation maximum peak. Other algorithms for sub-pixel interpolation were tested. The results of a simple comparative exercise are shown in Appendix 3.A.1.

### 3.4.3 Error metrics

All metrics are based on the error vector magnitude taken for the centre of each pixel in the OpF method, and for the centre of each IA, for PIV. If the error vector is defined as  $\mathbf{r} = \mathbf{u} - \mathbf{u}^{(g)}$ , where  $\mathbf{u}$  is the computed velocity and  $\mathbf{u}^{(g)}$  is the true velocity, its absolute magnitude is

$$\epsilon \equiv \|\mathbf{r}\| = \sqrt{(u - u^{(g)})^2 + (v - v^{(g)})^2}, \quad (3.1)$$



where  $u$  and  $v$  are the  $x$  and  $y$  velocity components, respectively, and  $u^{(g)}$  and  $v^{(g)}$  are the corresponding values. Equation (3.1) expresses the absolute error which may not be appropriate to evaluate the performance of the methods when displacements are small. The relative error is thus also computed:

$$\epsilon_R = \frac{\epsilon}{\|\mathbf{u}^{(g)}\|} \quad (3.2)$$

where  $\|\mathbf{u}^{(g)}\|$  is zero, the relative error vector is discarded.

To attain statistical significance, the errors are computed over a SIG. The absolute error of each SIG is

$$\epsilon_{SIG} = \frac{\sum_{f=1}^n \sum_{k=1}^U \sum_{l=1}^V \epsilon_{k,l}^{(f)}}{nUV}, \quad (3.3)$$

and the SIG relative error is

$$\epsilon_{R-SIG} = \frac{\sum_{f=1}^n \sum_{k=1}^U \sum_{l=1}^V \epsilon_{Rk,l}^{(f)}}{nUV}, \quad (3.4)$$

where the number of image pairs in a SIG is  $n = 10$  (section 3.3.2),  $U$  is 480 (vectors, corresponding to as many px) and  $V$  is also 480 (vectors or px). A band of 16 vectors wide is removed all around the margins of the image for OpF SIGs, in order to reduce boundary effects associated to non-physical boundaries. For PIV,  $U = 30$  (vectors or interrogation areas) and  $V = 30$ . Again, an IA margin of  $16 \text{ px} \times 16 \text{ px}$  is removed all around the image in order to mitigate non-physical boundary effects.

Both absolute errors ( $\epsilon_{SIG}$ ) and relative errors ( $\epsilon_{R-SIG}$ ) are expressed in decibels, respectively  $dBpx = 20\log_{10}(\epsilon_{SIG})$ , for absolute errors, and  $dB = 20\log_{10}(\epsilon_{R-SIG})$  for relative errors.

The error of sets of SIG ensembles is computed as

$$E = \frac{\sum_{g=1}^N 20 \log_{10}(\epsilon_{SIG_g})}{N}, \quad [dBpx] \quad (3.5)$$

and

$$E_R = \frac{\sum_{g=1}^N 20 \log_{10}(\epsilon_{R-SIG_g})}{N}, \quad [dB] \quad (3.6)$$

where  $N$  is the number of SIGs contributing to the average.

It should be noted that errors for OpF methods were also computed per interrogation area in order to be commensurable with PIV. This was achieved by first splitting an image into IAs of  $16 \text{ px} \times 16 \text{ px}$  that exactly overlap with the PIV IAs. At each IA, a single vector was taken as the average of the OpF vectors within the IA:

$$u_{IA} = \frac{\sum_{i=1}^{16} \sum_{j=1}^{16} u_{ij}}{16 \times 16}, \quad v_{IA} = \frac{\sum_{i=1}^{16} \sum_{j=1}^{16} v_{ij}}{16 \times 16}.$$

The same was done for the ground truth reference velocities in order to obtain a single vector per IA:

$$u_{IA}^{(g)} = \frac{\sum_{i=1}^{16} \sum_{j=1}^{16} u_{ij}^{(g)}}{16 \times 16}, \quad v_{IA}^{(g)} = \frac{\sum_{i=1}^{16} \sum_{j=1}^{16} v_{ij}^{(g)}}{16 \times 16}.$$

### 3.4.4 Quantification of intrinsic parameters of OpF methods and PIV image analysis

Most OpF methods rely on intrinsic parameters that require quantification [4]. Such quantification normally depends on imaging conditions and can be optimized. The optimization involved taking a sample pair of images from the benchmark image database and from a set of flows with  $N_i = 6$  and  $N_i = 1$  and  $N_i = 12$  and particle spot radius of 1.5 px (3.0 px diameter), the range for which PIV achieves the greatest accuracy (see Raffel et al. 38). The parameters of each OpF method are determined in order to attain the best accuracy for this range. It is possible to tune the algorithms to achieve better performance for specific cases but that was not the purpose of this quantification step. The focus was to obtain a reference set of intrinsic parameters for which OpF methods achieve a good (but not necessarily optimal) overall performance for the range of spot sizes and number particle concentration values for which PIV is known to have a good performance.

For the Horn-Schunck methods, the Lagrange multiplier is a function of the pyramidal level,  $l$ , of the image bit depth,  $b$ , and of the number density of particles,  $N_i$ ,  $\lambda(l, b, N_i)$ . The adopted values are shown in Table 3.3. The  $\lambda$  function was obtained by performing a non-exhaustive binary search on the individual  $\lambda$  values for a given tuple of  $l, b, N_i$  that minimized the average SIGs error over a selected set of flows and image pairs. As seen in Table 3.3 the Horn-Schunck method is sensitive to the image characteristics – the values of the Lagrange multiplier  $\lambda(l, b, N_i)$  vary from as low as 6 to as high as 1460.

Table 3.3: Values of the Lagrange multiplier ( $\lambda$ ) for the Horn-Schunck method as a function of pyramidal level ( $l$ ), image bit-depth ( $b$ ) and number of particles per IA volume ( $N_i$ ).

| $l$ | $b$ | $N_i$ | $\lambda(l, b, N_i)$ | $l$ | $b$ | $N_i$ | $\lambda(l, b, N_i)$ |
|-----|-----|-------|----------------------|-----|-----|-------|----------------------|
| 1   | 8   | 1     | 6                    | 2   | 8   | 1     | 40                   |
| 1   | 8   | 6     | 21                   | 2   | 8   | 6     | 45                   |
| 1   | 8   | 12    | 30                   | 2   | 8   | 12    | 61                   |
| 1   | 8   | 16    | 34                   | 2   | 8   | 16    | 75                   |
| 1   | 10  | 1     | 20                   | 2   | 10  | 1     | 450                  |
| 1   | 10  | 6     | 77                   | 2   | 10  | 6     | 450                  |
| 1   | 10  | 12    | 119                  | 2   | 10  | 12    | 450                  |
| 1   | 10  | 16    | 131                  | 2   | 10  | 16    | 500                  |
| 1   | 12  | 1     | 82                   | 2   | 12  | 1     | 500                  |
| 1   | 12  | 6     | 325                  | 2   | 12  | 6     | 920                  |
| 1   | 12  | 12    | 470                  | 2   | 12  | 12    | 1260                 |
| 1   | 12  | 16    | 550                  | 2   | 12  | 16    | 1460                 |

Several parameters are common to OpF methods, including the Liu-Shen method, when applicable:  $w$  is the window size,  $nI$  is the number of iterations,  $h$  is the Lagrange multiplier value for Liu-Shen method and  $MI$  is the maximum number of iterations for the same method. We configured the Horn-Schunck method with  $nI = 600$ . Horn-Schunck/Liu-Shen combination was setup with  $h = 5 \times 10^8$  and  $MI = 60$ . Lucas-Kanade was setup with  $w = 27$  and  $nI = 600$ . Additionally, when doing Lucas-Kanade/Liu-Shen combination, the Liu-Shen method was configured with  $h = 1 \times 10^{11}$  and  $MI = 60$ . The Farnebäck method was configured with  $w = 33$  and  $nI = 200$ . When doing Farnebäck/Liu-Shen combination, the Liu-Shen method was setup with  $h = 1 \times 10^{11}$  and  $MI = 60$ .

## 3.5 Results

### 3.5.1 Pyramidal vs non-pyramidal optical flow and bit-depths

There are several published works on multi-pass pyramidal methods, e.g., [41], [43] or [37], among others. The general consensus is that pyramidal multi-pass improves the accuracy of OpF methods. We have thus decided to benchmark only 2-level pyramidal versions of the OpF methods. An assessment of the impact of pyramidal 2-level on the results of the Horn-Schunck, Lucas-Kanade and Farnebäck methods is presented in the Appendix 3.A.2.

It was observed that the accuracy of all methods improved for its 2-level pyramidal version for 4.0 px displacements (25 % of the side of the PIV IA). For smaller displacements, no marked accuracy improve was registered but there were also no added problems such as excessive run time, poorly converging of iterative schemes, etc.. It was also found that some methods inherently perform better in a sub-set of the considered bit-depths and that the pyramidal processing is not more affected by the different bit-depths than the base algorithm. The most relevant example is the case of the Farnebäck algorithm whose pyramidal and base versions exhibit best accuracy only for 10- and 12-bit image depths.

All results in the remainder of this paper have been obtained with 2-level pyramidal versions of the considered OpF methods.

### 3.5.2 Combination with Liu-Shen (2008) and effect of image bit-depth

We have combined the algorithms of [4] (Horn-Schunck), [5] (Lucas-Kanade) and [7] (Farnebäck), herein the base algorithms, with the [25] (Liu-Shen) algorithm. The combination was embedded in the 2-level pyramidal scheme: the base algorithm was applied first to the image pair with a lower resolution; the resulting flow field was employed as the first estimate of the Liu-Shen iterative process, which concludes the first pass; the resulting flow field is then interpolated (bi-linear interpolation) to assign a vector to each pixel; the flow field obtained in the first pass is then used to deform the second image, in case of the Horn-Schunck method; in the case of the Lucas-Kanade and of the Farnebäck methods, the flow field from the first pass was used as a first estimate in their iterative processes; finally, the results of the base methods were employed as first estimates of the Liu-Shen method, which concluded the second pass of the 2-level pyramidal process.

Table 3.4 summarizes the effects on accuracy resulting from combining the three base algorithms with the Liu-Shen method. It includes the values of the average (absolute and relative) errors and corresponding standard deviations. All statistics have been computed from SIG errors, not errors of image pairs. Average errors are expressed in dB – low errors are associated to low dBpx values, i.e. negative numbers with high absolute value.

It is clear from Table 3.4 that, in general, combining the Liu-Shen method with any of the base algorithms improves accuracy. The improvements in accuracy, however, depend on image bit depth and may not be uniform. For instance, the Horn-Schunck algorithm improves from  $-1.46$  dBpx to  $-3.28$  dBpx, if the images are 8-bit, but only from  $1.04$  dBpx to  $-0.45$  dBpx for 10-bit images. This is further

explored in Figures 3.3 and 3.4.

Table 3.4: Absolute and relative errors (Avg.) and standard deviations (Std.) of the base algorithms and of the base algorithms combined with the Liu-Shen method. All statistics are discriminated by bit depth and are computed from SIG ensembles. Each SIG ensemble is composed of all flow types, particle spot sizes, particle concentrations and maximum displacements. All methods feature a pyramidal 2-level pass.

| Bit-depth | Algorithm             | SIG Ensemble<br>Absolute Errors<br>(dBpx) |       | SIG Ensemble<br>Relative Errors<br>(dB) |       |
|-----------|-----------------------|---|-------|---|-------|
|           |                       | Avg.                                      | Std.  | Avg.                                    | Std.  |
| 8         | Farnebäck             | -19.38                                    | 10.65 | -17.68                                  | 10.55 |
|           | Farnebäck/Liu-Shen    | -21.73                                    | 11.81 | -19.89                                  | 11.37 |
|           | Lucas-Kanade          | -21.90                                    | 11.61 | -19.52                                  | 13.07 |
|           | Lucas-Kanade/Liu-Shen | -25.02                                    | 11.22 | -22.39                                  | 12.79 |
|           | Horn-Schunck          | -11.46                                    | 9.45  | -10.61                                  | 7.94  |
|           | Horn-Schunck/Liu-Shen | -13.67                                    | 11.08 | -12.82                                  | 9.32  |
| 10        | Farnebäck             | -21.66                                    | 8.59  | -19.40                                  | 10.40 |
|           | Farnebäck/Liu-Shen    | -24.79                                    | 8.94  | -22.29                                  | 10.60 |
|           | Lucas-Kanade          | -21.92                                    | 11.61 | -19.55                                  | 13.07 |
|           | Lucas-Kanade/Liu-Shen | -25.05                                    | 11.22 | -22.42                                  | 12.78 |
|           | Horn-Schunck          | -11.17                                    | 9.36  | -10.38                                  | 7.55  |
|           | Horn-Schunck/Liu-Shen | -13.40                                    | 10.98 | -12.62                                  | 8.92  |
| 12        | Farnebäck             | -21.80                                    | 8.45  | -19.46                                  | 10.55 |
|           | Farnebäck/Liu-Shen    | -24.99                                    | 8.68  | -22.41                                  | 10.67 |
|           | Lucas-Kanade          | -21.93                                    | 11.62 | -19.56                                  | 13.08 |
|           | Lucas-Kanade/Liu-Shen | -25.05                                    | 11.24 | -22.43                                  | 12.80 |
|           | Horn-Schunck          | -11.41                                    | 9.39  | -10.56                                  | 7.83  |
|           | Horn-Schunck/Liu-Shen | -13.65                                    | 11.03 | -12.81                                  | 9.22  |

Figure 3.3 shows the distribution of errors of the base algorithms and respective combinations with Liu-Shen, for 8-bit images. In what concerns the Horn-Schunck algorithm, it is observed that combining it with the Liu-Shen method does not significantly affect large errors. Smaller errors, on the contrary, are further diminished, resulting in a wider distribution and, thus, increased standard variation (see also Table 3.4). The same trend is observed in the Farnebäck algorithm – combination with the Liu-Shen method does not contribute to eliminate large errors. The improvement in the Farnebäck results is observed in the lower error range, which, again, results in the increase of the standard deviation of the distribution. On the contrary, combining the Lucas-Kanade algorithm with the Liu-Shen method does result in an increase of accuracy across the entire error distribution. The absolute errors of the Lucas-Kanade algorithm are mostly grouped around the norm  $-30$  dBpx, with a thin tail of larger errors. Combining the Lucas-Kanade algorithm with the Liu-Shen method eliminates the errors larger than 1 dBpx and pushes the norm down to about  $-32$  dBpx. The resulting distribution of Lucas-Kanade errors preserves the standard variation and has a lower mean (see also Table 3.4). The same interpretation can be derived, in this case, from the relative errors.

It is clear from Table 3.4 that the conclusions drawn from 8-bit images remain valid for 10-bit or 12-bit images only for the Horn-Schunck and Lucas-Kanade algorithms – the errors and standard deviations for these methods are very similar for 8-, 10- or 12-bit image depths. For the Farnebäck algorithm, working with 10-bit or 12-bit images results in different error distributions, even if combining with the

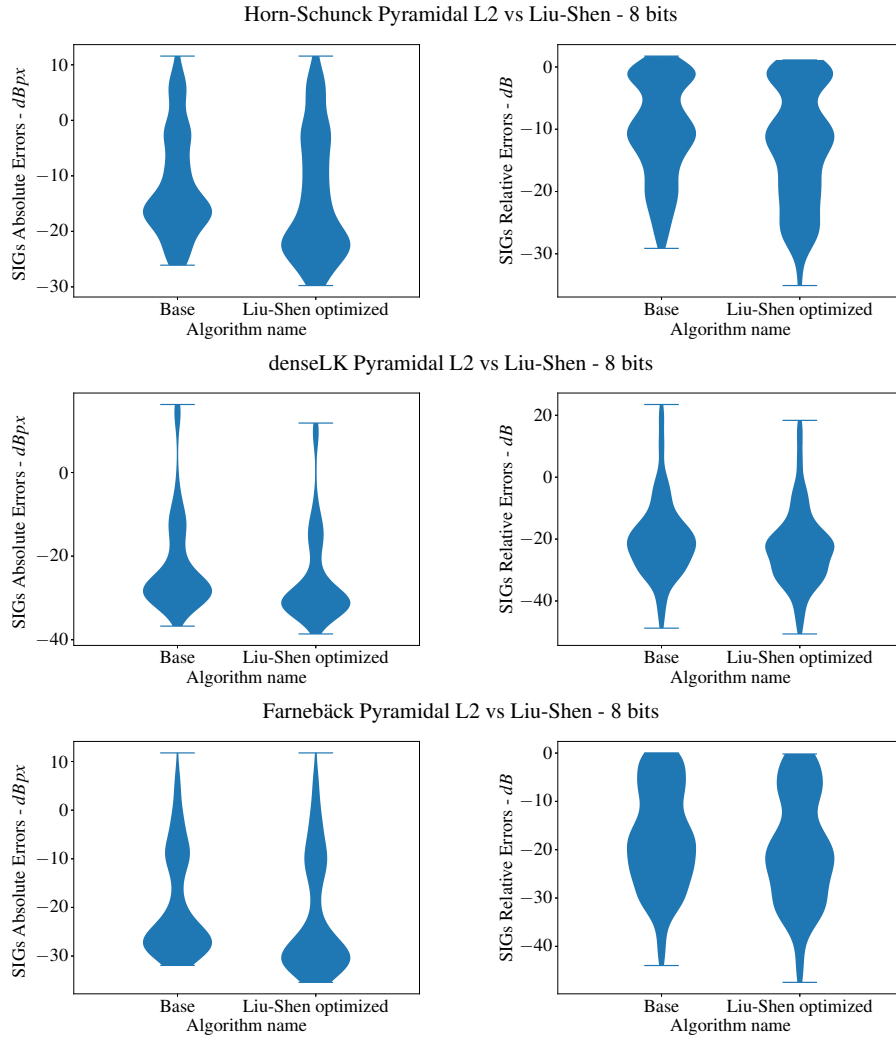


Figure 3.3: Distribution of absolute (left) and relative (right) errors for the base algorithms and for the base algorithms combined with the Liu-Shen method, 8-bit images. Top row: Horn-Schunck; middle row: Lucas-Kanade; bottom row: Farnebäck. All implementations comprise a pyramidal 2-level pass.

Liu-Shen method still increases accuracy.

Figure 3.4 allows for a more complete characterization of the Farnebäck/Liu-Shen combination. Comparing figures 3.3 (bottom row) and 3.4 reveals that: i) for 8-bit images, the (relative or absolute) error distribution of the Farnebäck method features large errors (larger than  $-5$  dBpx, in the absolute case) that essentially disappear for 12-bit images; ii) in the range  $-20$  to  $-5$  dBpx, the absolute error is significantly reduced in the case of 12-bit, in comparison to the 8-bit images; iii) combining the Farnebäck algorithm with the Liu-Shen method allows for the elimination of very large errors (larger than  $-3$  dBpx, in the case of absolute errors) while reducing the errors throughout the entire range. As seen in Table 3.4, the most significant error reduction for the Farnebäck algorithm results from employing 10-bit images instead of 8-bit. Further reduction is attained by employing 12-bit images. Combining with the Liu-Shen method improves the accuracy of the base Farnebäck algorithm for all image bit depths. Combination with Liu-Shen is more effective in reducing the errors of the FB algorithm at 10- or 12-bits, resulting in a slightly larger standard deviation. While this is the general trend, it should be noticed that some outliers have been observed. For instance, for uniform flow, at  $4.0px$  maximum displacement and  $3.0px$  particle spot size and  $N_i = 1$ , a small increase ( $0.03$  dBpx) of both absolute and relative errors have been observed.

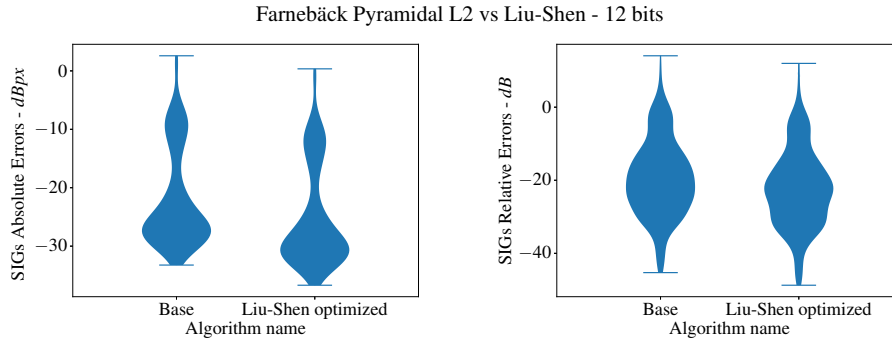


Figure 3.4: Distribution of absolute (left) and relative (right) errors for the Farnebäck algorithm and for the same algorithm combined with the Liu-Shen method, 12-bit images. The implementation comprises a pyramidal 2-level pass.

The analysis of relative errors provides essentially the same conclusions as that of absolute errors. In some cases of very low particle concentrations with  $N_i = 1$  and small particle spot sizes ( $1.0$  px diameter) combining the Farnebäck algorithm with the Liu-Shen method does not bring about relative error improvements. This may indicate that combination with Liu-Shen reduces errors everywhere except near the walls or singularities where the true velocity is very small.

As a summary, we point out that all base algorithms benefit from combination with the [25] method (Liu-Shen). In the reminder of the paper, all further analysis is carried out with Liu-Shen combined versions of the base algorithms. We underline that there no reason to use 10-bit or 12-bit image depths when applying the [4] or the [5] algorithms (combined or not with Liu-Shen): the latter does not show significant improvements and the former can even exhibit a lower accuracy. In case of the [7] algorithm, its accuracy greatly improves for images with 10- or 12-bit depths.

### 3.5.3 Performance of OpF methods according to flow type

We quantify the relative performance of the OpF combinations Horn-Schunck/ Liu-Shen, Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen. All implementations feature a 2-level pyramidal pass. We have narrowed the analysis down to these three combinations of methods since, as seen in the previous sections, non-pyramidal implementations and base algorithms without combination with Liu-Shen result in larger errors.

We assess performance exclusively by comparing absolute and relative errors to ground-truth. In this section, the metric used to compare the performance of any two methods,  $A$  and  $B$ , is

$$\mathcal{E} = E^{(A)} - E^{(B)} \quad (3.7)$$

where  $\mathcal{E}$  is the comparative error and  $E^{(A)}$  and  $E^{(B)}$  are the errors of methods  $A$  and  $B$ , respectively, given by equations (3.5) or (3.6) (thus computed over sets of SIGs). The comparative analysis discriminates the type of flow and the image bit depth. To compute the set errors  $E^{(A)}$  and  $E^{(B)}$ , all other image and imaging characteristics are lumped.

The results are shown in the form of box-plots – Figures 3.5 and 3.6. Each box, marks the range limited by the end of the Q1 quartile and the the start of Q3 quartile. The vertical red line inside the box marks the Q2 quartile, *i.e.* the median. The small circles represent outliers, with values outside the band  $3IQR$  centered around the mean of the distribution, where  $IQR$  is the length of the box, defined as the difference between quartiles 3 and 1.

Equation (3.7) returns a positive value when method  $B$  has lower errors than method  $A$  (for which we say method  $B$  performs better than method  $A$ ) and the boxes in the box-plot lean towards the half space of method  $B$ . In general, the boxes lean towards the half-space of the better performing method. For instance, Figure 3.5, top row, reveals that the Lucas-Kanade/Liu-Shen combination generally performs better than the Horn-Schunck/Liu-Shen combination.

Figure 3.5, top row, shows the comparison of Horn-Shunck/Liu-Shen with Lucas-Kanade/Liu-Shen. The Lucas-Kanade/Liu-Shen combination exhibits a better performance for all flows. The proportion of SIGs for which the Horn-Shunck/Liu-Shen combination has smaller absolute errors is less than 10 % in all cases and less than 1 % in the case of the Poiseuille flow. However, observing the relative errors, it is in the Poiseuille flow that the Lucas-Kanade/Liu-Shen combination offers the less expressive improvement over the Horn-Shunck/Liu-Shen. The Poiseuille is a viscous boundary layer flow for which the velocity is zero at the walls, where the velocity gradients are also large. This case has the highest proportion of pixels with near-zero velocities, relatively to the other flows. The stagnation point flow is inviscid and the vortex flows are not bound, the flow near singularities where the flow velocity is zero does not occupy large proportions of the region of interest. Hence, we interpret the results of Figure 3.5(top row) as an indication that the Lucas-Kanade/Liu-Shen combination does, in general, perform better everywhere except near walls where the no-slip condition applies. In these case, its improvements relatively to the Horn-Shunck/Liu-Shen combination, are less expressive. A closer inspection of the overall results reveals that Horn-Schunck/Liu-Shen combination has better accuracy, in terms of both

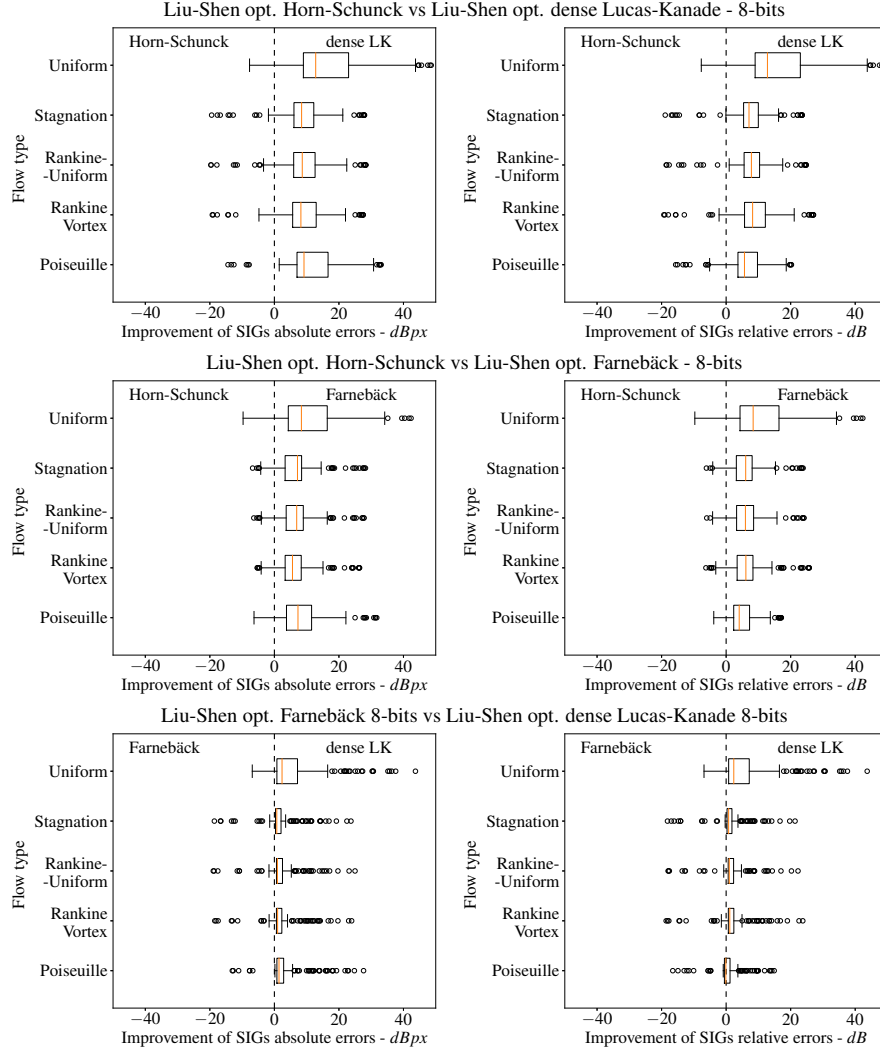


Figure 3.5: Direct comparison of OpF combinations for 8-bit images in terms of absolute errors (left column) and relative errors (right column). Top row shows the comparison between Horn-Shunck/Liu-Shen and Lucas-Kanade/Liu-Shen. Middle row shows the comparison between Horn-Shunck/Liu-Shen and Farnebäck/Liu-Shen. Bottom row shows the comparison between Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen. All methods include a pyramidal 2-level scheme. Boxplots condense the distribution of comparison errors, defined by equation (3.7). Each box encompasses the second and third quartiles. The whiskers encompass a band of  $3IQR$  centered around the mean of the distribution. Circles represent values lying beyond this band. The median of the distribution is represented by the red bar in the box.



relative and absolute errors, for images with particle spot sizes of 1 pixel and  $N_i = 1$ . These imaging conditions are avoided in Fluid Mechanics applications. Other outliers have been found for images with particle spot sizes of 1 pixel,  $N_i = 6$ , high noise levels (15 dB WGIN) and low displacements 0.8 px.

Figure 3.5(middle row) compares the Horn-Shunck/Liu-Shen and the Farne-bäck/Liu-Shen combinations. The Farne-bäck/Liu-Shen combination outperforms the Horn-Shunck/Liu-Shen combination in all flow cases. As in the previous case, the improvement of the Farne-bäck/Liu-Shen combination is less relevant in the case of the Poiseuille flow. Near-wall viscous flows, with low velocities and strong gradients, are the type of flows where improvements to Horn-Shunck/Liu-Shen combination are harder to achieve. It should be noticed, in respect to the overall results, that the Horn-Schunck/Liu-Shen combination performs better, at the 8-bit image depths, for a small number of outliers with particle spot sizes of 1.0 px, 2.0 px or 3.0 px and  $N_i = 1$ . Again, such an imaging condition is not desirable in real applications because of the low tracer concentration.

The Lucas-Kanade/Liu-Shen and Farne-bäck/Liu-Shen combinations are compared in Figure 3.5(bottom row). The results for flows with strong vortical cores (Rankine vortex and Rankine vortex with superimposed uniform flow) and for the irrotational flow (stagnation point) are the same: the Lucas-Kanade/Liu-Shen combination is marginally better while only a few outliers are favorable to the Farne-bäck/Liu-Shen combination. Hence, the type of flow – rotational or deformation dominated – is not relevant to discriminate between these OpF methods whose overall performance is very similar. The Lucas-Kanade/Liu-Shen combination performs better than the Farne-bäck/Liu-Shen combination if the flow is uniform, but this is the least interesting flow situation. In the case of the Poiseuille flow, the absolute errors of the Lucas-Kanade/Liu-Shen combination are marginally smaller. However, the relative errors of the Lucas-Kanade/Liu-Shen combination are higher than those of the Farne-bäck/Liu-Shen combination. This indicates that the Lucas-Kanade/Liu-Shen combination only improves the Farne-bäck/Liu-Shen combination in the core of the flow, not near the strong-gradient, zero-velocity walls.

The comparison between the Farne-bäck/Liu-Shen and Lucas-Kanade/Liu-Shen combinations for 10-bit image depths is assessed in Figure 3.6. It was seen in section 3.5.2, when comparing Figure 3.3(bottom row) with Figure 3.4, that the Farne-bäck/Liu-Shen combination performs better at 10-bit image depths. Figure 3.6 shows that the performance of both combinations is thus nearly identical for this image bit depth. The Lucas-Kanade/Liu-Shen combination is still slightly better in the case of the uniform flow. In the case of the Poiseuille flow, although the absolute errors of the Farne-bäck/Liu-Shen combination are slightly larger, its performance is better near the walls subjected to the no-slip condition.

This comparative analysis reveals that the OpF methods most suited to be applied in Fluid Mechanics, for instance to re-use old PIV databases and increasing vector density, are the Lucas-Kanade/Liu-Shen and the Farne-bäck/Liu-Shen combinations. The former, Lucas-Kanade/Liu-Shen, performs well in 8-bit image depths, which is the most likely format for old PIV databases, and also in 10-bit or higher bit depths. The Farne-bäck/Liu-Shen combination may be a good option for viscous flows in the near vicinity of walls but, in general, achieves highest performance for 10- or higher bit depths. The Horn-Schunck/Liu-Shen combination shows a smaller overall accuracy. The optimal value of the Lagrange multiplier is difficult to determine, given that it is very sensitive to the image characteristics, which may

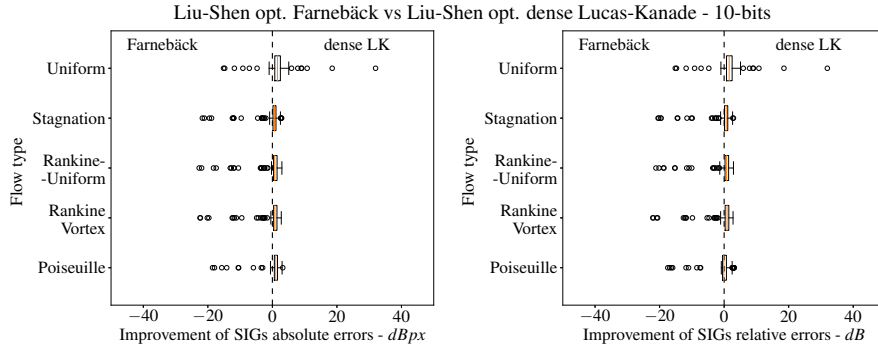


Figure 3.6: Direct comparison of the Farnebäck/Liu-Shen and Lucas-Kanade/Liu-Shen combinations for 8-bit images in terms of absolute errors (left plot) and relative errors (right plot). Both methods include a pyramidal 2-level scheme. Box-plots condense the distribution of comparison errors, defined by equation (3.7) Reminder of caption as in Figure 3.5.

explain some loss of performance.

### 3.5.4 Influence of imaged particle spot sizes and particle number density

In this section we evaluate the influence of particle spot sizes on accuracy, for the relevant OpF methods. We also include the results of our implementation of PIV, which provide a reference scenario.

Figure 3.7 allows for a discussion of the accuracy of the Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen combinations and of PIV for the Rankine vortex with superimposed uniform flow. The plots show the errors of individual SIGs, with a total of 48 SIGs per bit-depth and per method. Each SIG is characterized by its particle concentration per IA, its spot size and WGIN noise value. The maximum displacement is the same for all SIGs - 25 % of the size of the IA, 4 px. The errors (relative) are plotted as a function of the number tracer particle concentration and parameterized for the image spot size, and noise level.

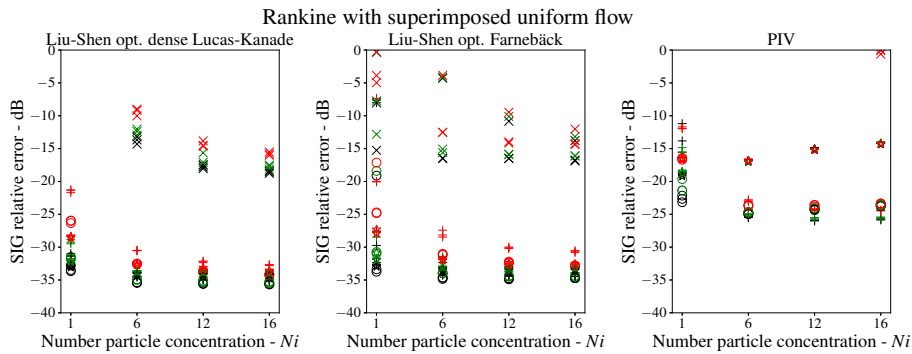


Figure 3.7: Accuracy vs particle spot size, WGIN and  $N_i$  for Rankine vortex with superimposed uniform flow at 4.0 px max. displacement, for all 8, 10 and 12-bit image depths - Results shown for Lucas-Kanade/Liu-Shen combination, Farnebäck/Liu-Shen combination and PIV (Legend:  $\times$  - denotes 1 px diameter imaged particle spot size,  $+$  - denotes 2 px diameter imaged particle spot size,  $\bigcirc$  - denotes 3 px diameter imaged particle spot size,  $*$  - denotes 6 px diameter imaged particle spot size, Black - denotes 0 dBW WGIN, Green - denotes 5 dBW WGIN, Red - denotes 15 dBW WGIN).

Our implementation of PIV is not optimized. It is multi-pass, starting at a square IA size of 128 px and ending at 16 px but does not perform Gaussian window weighting, window deformation or bi-cubic intensity interpolation. However, our PIV implementation does capture essential features: as seen in

Figure 3.7(right), the lowest errors are obtained for particle number concentrations  $N_i \geq 6$  and tracer spot sizes between 2 px and 3 px. These are well-known results for double-pulsed images with Gaussian correlation peak approximators [38, 65–67]. [67] showed that PIV precision is determined by the tracer spot size, whereas measurement reliability is determined by  $N_i$ . The errors in Figure 3.7(right) can be discussed in this framework – precision increases from 1 px (not shown in the plot since the errors are above 1 dB) and to 2 px and diminishes again for particles larger than 3 px. For small particles (1-3 px), an increase in  $N_i$  represents an increase in accuracy due to higher reliability but only for  $N_i < 6$ . In the optimal precision range of particle spot sizes (2-3 px), increasing the value of  $N_i$  beyond 6 particles does not bring about an increase of accuracy only an improvement in the signal-to-noise ratio. If the spot size is large (6 px) an increase in  $N_i$  results in improving accuracy for  $N_i < 6$ . For  $N_i \geq 6$ , the errors increase as the 6 px spot sized imaged particles coalesce and form speckles, thus stepping outside the PIV realm.

The distribution of relative errors for the tested OpF combinations, Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen, are shown in Figure 3.7(left) and Figure 3.7(centre), respectively. There are some similarities to PIV: the methods perform better for spot sizes larger than 1 px and for particle concentrations  $N_i \geq 6$ . There are important differences too: unlike PIV, the OpF methods perform well for large spot sizes (6 px). In fact, the relative errors of both methods are slightly better for 6 px than for 2 px spot sizes. The accuracy of both methods seems to increase with  $N_i$ , especially if noise levels are high and independently of spot size. This may indicate that OpF methods may work well outside the PIV realm and into the speckle velocimetry domain. It is also possible to use dense Lucas-Kanade or Farnebäck methods for extremely low particle concentrations (even  $N_i = 1$ ) provided that noise is low.

The influence of particle spot size and tracer number density on accuracy has been assessed experimentally and theoretically for correlation-based PIV. Optical flow methods do not have a common theoretical basis and its accuracy requires further theoretical investment. The results produced in this section show that OpF methods perform well in the range where correlation-based PIV is optimal. However, the range of  $N_i$  and  $d_\tau$  values for which OpF methods perform well seems wider. These results may be contextualized in future theoretical studies.

### 3.6 Velocity field analysis

So far we have addressed the accuracy of OpF methods with statistical tools and a lumped analysis – errors have been quantified for the entire region of interest of each flow (equations 3.3 and 3.4). We now address the issue of the quality of the description of the flow field, in particular the verification of continuity.

All flows described in Table 3.1 are planar and solenoidal in two-dimensions. If the difference between the velocity ground truth and the estimated velocity is  $\mathbf{r} = \mathbf{u} - \mathbf{u}^{(g)}$ , where  $\mathbf{u}$  is the estimated velocity vector and  $\mathbf{u}^{(g)}$  is the true velocity vector, then the divergence of the velocity vector becomes

$$\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{r} = \lim_{\delta x_j \rightarrow 0} \frac{r_j(x_j + \delta x_j) - r_j(x_j)}{\delta x_j} \quad (3.8)$$

(index  $j$  indicates spatial directions and its repetition indicates summation in this mixed notation). We note that  $r_j(x_j + \delta x_j) - r_j(x_j) \leq \epsilon(\mathbf{x})$ ,  $\forall j = 1, 2$ , where  $\epsilon$  is computed by equation (3.1). Hence, ensemble-averaging over a SIG, one has  $\sum_{j=1}^2 \left( \overline{r_j(x_j + \delta x_j) - r_j(x_j)} \right) = \mathcal{O}(\bar{\epsilon}(\mathbf{x}))$ . Introducing this result in equation (3.8) and performing a first-order approximation to the derivative, one has

$$\nabla \cdot \bar{\mathbf{r}} \approx \frac{\bar{r}_j(x_j + \delta x_j) - \bar{r}_j(x_j)}{\delta x_j} = \mathcal{O}\left(\frac{\bar{\epsilon}}{\delta x}\right). \quad (3.9)$$

with  $\delta x = \delta x_j$ ,  $\forall j$ , for square pixels. In section 3.5.2 we have indicated that the average errors of the Lucas-Kanade/Liu-Shen combinations are as low as  $-25$  db, for all imaging conditions and image characteristics. For adequate tracer imaging conditions ( $N_i \geq 6$  and  $2 \leq d_\tau < 6$  px) absolute errors are expected to be lower. Considering  $\delta x = 1$  px for OpF methods, one has  $\nabla \cdot \bar{\mathbf{u}} = \nabla \cdot \bar{\mathbf{r}} = \mathcal{O}(10^{-2})$  to  $\mathcal{O}(10^{-1})$  pixel per time between consecutive images and per pixel. At the lower bound, errors of about 1 % of a pixel between two consecutive images should be enough to reproduce incompressibility.

The impact of absolute velocity errors depends evidently on the magnitude of the velocity vector itself. Hence, it may be instructive to cast the mean flow divergence as a function of relative errors. At any point in a given flow field, considering equation (3.2), one has  $\bar{\epsilon}_R(\mathbf{x}) = \bar{\epsilon}(\mathbf{x})/\|\mathbf{u}^{(g)}(\mathbf{x})\|$ , and thus

$$\nabla \cdot \bar{\mathbf{u}} = \nabla \cdot \bar{\mathbf{r}} = \mathcal{O}\left(\bar{\epsilon}_R(\mathbf{x}) \frac{\|\mathbf{u}^{(g)}(\mathbf{x})\|}{\delta x}\right). \quad (3.10)$$

In section 3.5.4 we have shown that for adequate tracer imaging conditions the average relative errors are  $\bar{\epsilon}_R(\mathbf{x}) < \mathcal{O}(10^{-2})$ . The magnitude of the true velocity is limited to 4 px per time interval between two consecutive frames ( $\frac{\|\mathbf{u}^{(g)}(\mathbf{x})\|}{\delta x} \leq 4$ ).

The relative errors are unevenly distributed in the region of interest – they are generally higher near the walls of viscous flows, where the velocity is lower, or near singularities. Hence, while it might be stated that, on average, the solenoidal condition is fulfilled by OpF Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen combinations in the range of  $d_\tau$  and  $N_i$  for which best accuracy is attained (see section 3.5.4), more detailed observations are needed in the regions where flow velocities are lower. To address this issue, we assess the quality the flow field per se, not as a statistical ensemble. We believe that the simplest way to evaluate continuity is to show the pattern of streamlines, showing that they do not cross or plunge into the boundary, remain close to the boundary if initiated near the boundary, and do not jitter appreciably.

We show the streamlines of three types of flow: the stagnation point (Figure 3.8), the Poiseuille flow (Figure 3.9) and a Rankine vortex with superimposed uniform flow (Figure 3.10). Unlike the uniform flow, these are complex flows since the region of interest features both large and small displacements. Velocities can be zero, near the walls (Poiseuille flow) or near the stagnation point. In the case of the Rankine vortex with superimposed uniform flow, the rotational core is not distributed around the point with zero velocity.

The imaging and synthetic flow characteristics were: maximum displacements of 4.0 px, corresponding to  $1/4$  of the PIV IA,  $d_\tau = 3.0$  px,  $N_i = 6$ ,  $\sigma_z = 0.025$  px and  $WGIN = 0$  dBW. In Figures 3.8, 3.9 and 3.10 we plot the streamlines of the velocity field obtained by the Lucas-Kanade/Liu-Shen combina-

tion, which was shown to have low relative and absolute errors in this imaging range, in two versions: the original flow field (1 px resolution) and a re-sampled flow field to match the lower resolution of the correlation-based PIV (16 px).

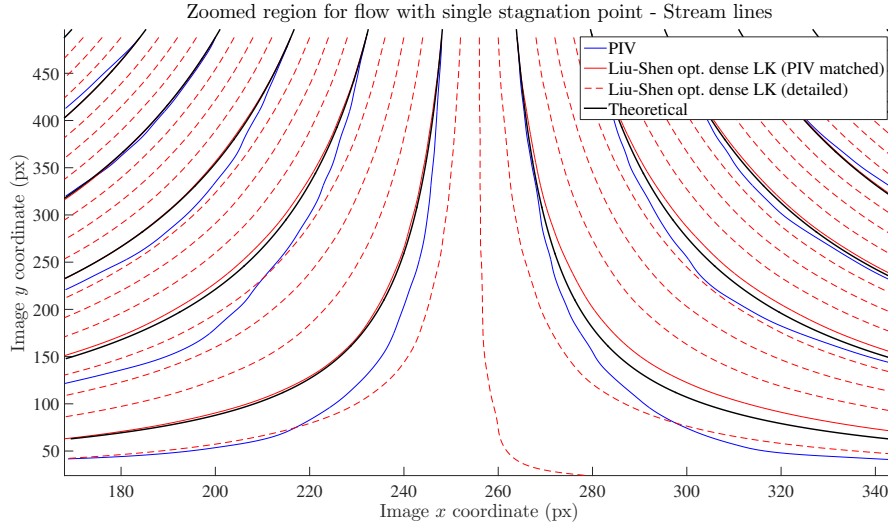


Figure 3.8: Streamlines for the flow with a single stagnation point, located at  $x = 250$  px,  $y = 0$  px (see also Figure 3.1). Flow direction from top to bottom. The parameters for image generation were: maximum displacement equal to 4.0 px, WGIN set to 0 dbW, diameter of the imaged particle spot equal to 3.0 px, particle number concentration per IA equal to 6, out-of-plane standard deviation equal to 0.025 px. Results are shown for the theoretical (black thick line), correlation-based PIV (blue thin line), full resolution Lucas-Kanade/Liu-Shen combination (red dashed line) and subsampled Lucas-Kanade/Liu-Shen combination (red thin line), to match the PIV resolution. Boundary conditions to all streamlines are located at  $y = 500$  px.

In the stagnation point flow, shown in Figure 3.8, the convective accelerations are large, leading to strongly curved streamlines. The full resolution OpF streamlines deviate slightly from the theoretical lines but only in the close vicinity of the stagnation point. The lower resolution OpF and the PIV streamlines deviate mildly from the theoretical streamlines, which is not unexpected given that the initial conditions are far from the stagnation point. It is worth noticing that the OpF streamlines do not exhibit jitter and that its subsampled version seems more accurate than the correlation-based PIV.

The analysis of the streamlines for the Poiseuille flow, shown in Figure 3.9, complements the considerations woven in section 3.5.2. It was seen that the Lucas-Kanade/Liu-Shen combination is more accurate than any other method for the Poiseuille flow, when absolute errors are concerned. The assessment of relative errors revealed that the accuracy was indeed very high in the outer flow region (center of the streamtube) but not so high in the inner flow region (near the wall). This is illustrated in Figure 3.9, where the OpF streamlines are jittery even for the highest resolution. All optical methods tend to underperform when the flow is characterized by large velocity gradients. In this case, it is clear that the dense velocity field of Lucas-Kanade/Liu-Shen combination is an improvement over the less resolved PIV description but may need further theoretical and implementation improvements to accurately describe the flow of viscous flows near solid walls.

The Rankine vortex with superimposed uniform flow (Figure 3.10) shows that the only major loss of accuracy of the OpF method occurs around the saddle point. Increased resolution results in a better

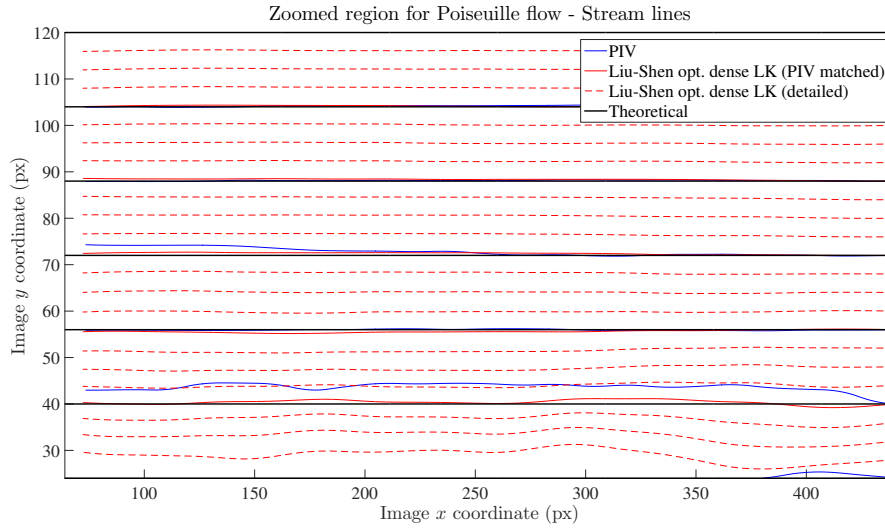


Figure 3.9: Streamlines for the Poiseuille flow. Flow direction from the right to the left. Remaining caption as in Figure 3.8.

flow description but, as is common in optical methods, errors are higher when streamline curvature is high.

It should be noted that the rotational core of this flow occurs at the centre of the region of interest ( $x = 250$  px,  $y = 250$  px in Figure 3.10, see also Figure 3.1). In this central region, OpF streamlines follow closely the theoretical curves, which indicates that the quality of the results does not depend on whether the flow is rotation-dominated or deformation-dominated but on velocity gradients and streamline curvature. It should be remembered that large curvatures of flow lines can occur for irrotational flows, as is the case of the flow near the saddle point in Figure 3.10 which lies outside the rotational flow core.

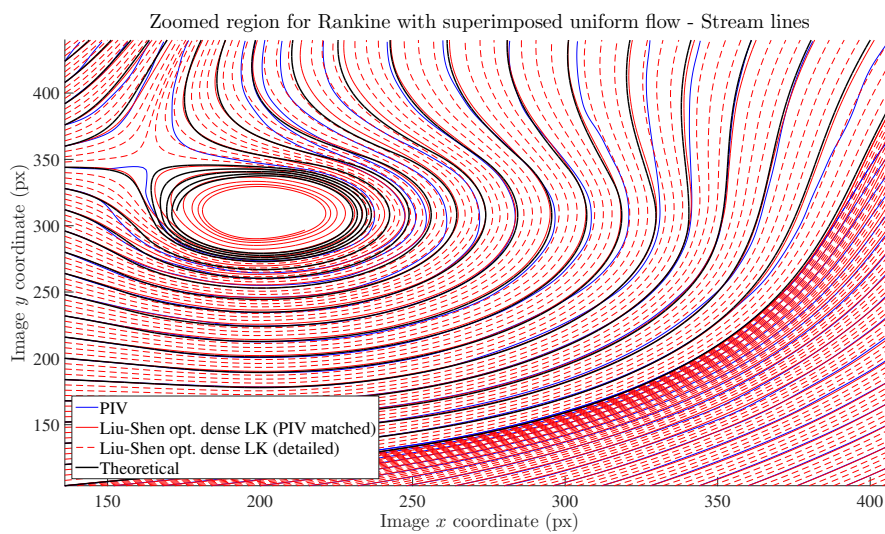


Figure 3.10: Streamlines for the Rankine Vortex with superimposed uniform flow. The vortex flow is direct (anti-clockwise) while the uniform flow is from bottom left to upper-right at  $45^\circ$ . Remaining caption as in Figure 3.8.

### 3.7 Discussion

The Lucas-Kanade/Liu-Shen and the Farnebäck/Liu-Shen combinations were shown to have the highest accuracy across a wide range of imaging conditions, image characteristics and flow types. Most of the outliers occurred for imaging conditions that are rare or undesired in PIV databases, namely very small tracer concentrations (e.g.  $N_i = 1$ ) or very large imaged particle spot size ( $d_\tau \geq 6$  px), as seen in section 3.5.3.

We now perform a final comparative accuracy assessment restricted to the imaging conditions and image characteristics for which OpF methods have higher accuracy. This optimal range largely coincides with the usual features of PIV databases for fluid mechanics:  $2.0 \leq d_\tau \leq 6.0$  [px],  $6 \leq N_i \leq 12$  and maximum displacements equal to 25% of the size of PIV IAs (see section 3.5.4). The Lucas-Kanade/Liu-Shen and Horn-Schunck/Liu-Shen were shown to perform better for 8-bit depth images while the Farnebäck/Liu-Shen combination was more accurate for 10-bit depth images; only these image characteristics are included in the comparative analysis.

The distribution of absolute errors in the conditions described above are shown in Figure 3.11 in the form of dot-and-whisker plots. The dot represents the median while the whiskers encompass the central part of the distribution, quartiles 2 and 3. It is clear that the Lucas-Kanade/Liu-Shen and the Farnebäck/Liu-Shen combinations have higher accuracies than the Horn-Schunck/Liu-Shen combination. It is also apparent that the Lucas-Kanade/Liu-Shen performs slightly better than the Farnebäck/Liu-Shen.

The lowest errors are obtained for the simple uniform flow, which is unsurprising. The Horn-Schunck/Liu-Shen combination features a large dispersion of errors in such a simple flow. This illustrates the difficulties in calibrating the parameters of the Horn-Schunck algorithm when the images are contaminated with noise.

The Lucas-Kanade/Liu-Shen and the Farnebäck/Liu-Shen combination feature higher relative errors for the Poiseuille flow. As seen in section 3.6, all optical methods show difficulties in the inner region of viscous flows due to the large velocity gradients associated to the no-slip condition. This may generate difficulties in fulfilling the continuity equation, as seen in section 3.6.

The stagnation point and the Rankine vortex with superimposed uniform flow generate errors of the same order of magnitude and, in both cases, mostly occurring in the regions with strong convective accelerations (near the stagnation point or around the saddle point in the latter flow).

The Rankine vortex flow generates large curvatures near its center, but the affected area represents a relatively small proportion of the region of interest. This justifies the relatively better performance of the methods in this flow.

It is thus apparent that large streamline curvatures and large velocity gradients influence most the accuracy of the OpF in the conditions of this benchmark. The results do not seem to be influenced by the nature of the flow – rotation- or deformation-dominated. Optical flow methods derived from the Lucas-Kanade method have been used in several fluid mechanics applications [23, 24, 49]; the “dense” version of the method can be considered well-known for researchers working on fluid mechanics applications.

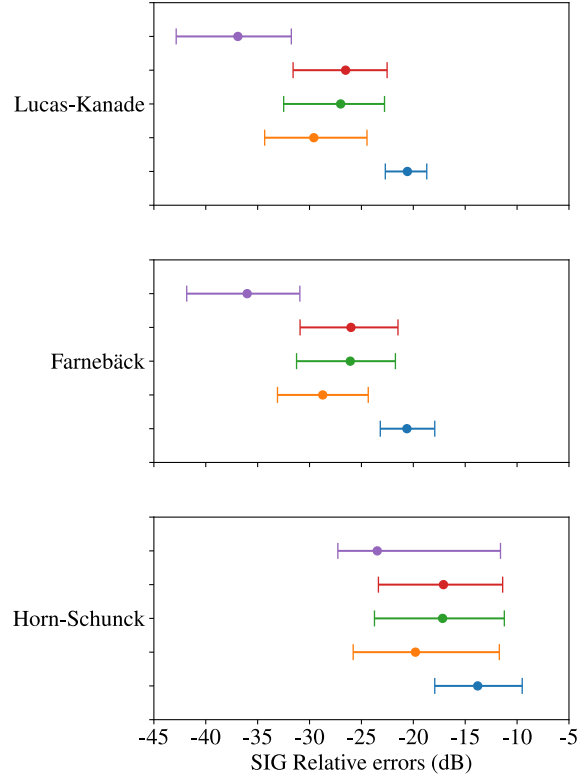


Figure 3.11: Distribution of relative errors expressing the accuracy of OpF methods per flow type in their optimal imaging conditions:  $2.0 \leq d_\tau \leq 6.0$  [px],  $6 \leq N_i \leq 12$  and 4 px maximum displacement. The assessment of accuracy is based on the empirical probability density of SIG errors. Each SIG error is computed by equation 3.3. The errors of the Lucas-Kanade/Liu-Shen and Horn-Schunck/Liu-Shen combinations were computed from 8-bit depth images while the errors of the Farnebäck/Liu-Shen combination were computed from 10-bit depth images. All methods featured a pyramidal 2-level pass. For each flow type and OpF method, the plot shows the central 50% of the error distribution (the width of the whiskers), centered around the median (the dot). Color code: — (Purple) for Uniform flow; — (Red) for the flow with a stagnation point; — (Green) for the Rankine vortex with superimposed Uniform flow; — (Orange) for the Rankine vortex; — (Blue) for the planar Poiseuille flow.



The same applies for the Liu-Shen method [25, 32, 33] and Horn-Schunk method. On the contrary, the Farnebäck method is relatively unknown to researchers in hydraulics and fluid mechanics. In fact, a search in the Scopus database (in August 2021), including Engineering, Mathematics, Physics and Astronomy and Earth and Planetary Sciences, revealed only 6 papers citing Farnebäck (2003) in the scope of hydraulics or fluid mechanics. None of these papers actively used or introduced novel aspects in the method. In the context of machine vision, Farnebäck (2003) was cited 1208 times. In contrast, Liu and Shen (2008) work was cited 155 times in Scopus (August 2021), all of them related to fluid mechanics applications.

We thus believe that the results of this benchmark support the view that it should be rewarding to invest further efforts in applying the Farnebäck method to fluid mechanics. This especially true since 10-bit or 12-bit images are currently standard in digital video. It is surely relevant to continue to exploit the Lucas-Kanade and the Liu-Shen methods, given that their combination yielded highly accurate results.

Hybridization of correlation-based PIV and Opf has been proposed by several researchers [28, 29, 33]. We believe it is promising to invest in hybrid methods involving the Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen combinations, the latter only for 10-bit or higher image depths, with special emphasis on tackling inaccuracies in the inner region of viscous flows.

To re-process old PIV databases, mostly acquired at 8-bit depth images, the results of this benchmark show that Lucas-Kanade/Liu-Shen is a good option, given that its accuracy is not affected by the image bit-depth. Moreover, OpF methods show a wider region of high accuracy in the parametric space of imaging conditions and image characteristics, relatively to PIV. This indicates that re-processing old PIV databases with low tracer concentrations, uneven seeding concentrations or inadequate (for correlation PIV) particle spot sizes may still be employed to extract valid data if an appropriate OpF method is used (with or without hybridization). The results of this benchmark indicate that the appropriate methods are the Lucas-Kanade/Liu-Shen or Farnebäck/Liu-Shen combinations, the latter for 10-bit images or higher.

### 3.8 Conclusion

We have carried out a benchmark study of optical flow (OpF) methods, including those most used in fluid mechanics applications – [5], [4] and [25] –, and the less used [7]. Considering the diversity of OpF approaches, we set up a benchmark exercise to assess the performance of these OpF algorithms across a parametric space of combinations of flow type, of imaging conditions and of digital image characteristics. Since velocity gradients can be divided in to a rate of rotation and a rate of deformation and most non-uniform flows are a combination of both, the cases included in the benchmark represented typical cases of rotation-dominated flows and deformation-dominated flows. As imaging conditions we included imaged particle features, laser characteristics and motion-related noise. As image characteristics we considered only bit depth and white noise. All databases were constituted by synthetic images.

The [25] “physics-based” optical flow was used only in combination with [5], [4] and [7]. The three latter methods were evaluated with and without a combination step with [25].

To evaluate the performance of all OpF methods, we computed the differences between the ground-

truth velocities and the estimated velocities. The errors of standard PIV solutions, for reference, and benchmarked OpF methods, were computed for a large number of combinations of deformation-dominated, rotation-dominated and uniform flows and different displacements, noise power, pixel particle spot size, image bit-depth and particle concentration. Both absolute and relative errors were considered to express accuracy. Comparison of both types of errors allows for insights on the difficulties of OpF methods in the inner region of flows of viscous fluids.

The main conclusions are summarized below.

- The accuracy of all base methods [4, 5, 7] increases when combined with the [25] method.
- The Lucas-Kanade/Liu-Shen and the Farnebäck/Liu-Shen combinations are, in general, the OpF methods with the best performance, i.e featuring smaller relative and absolute errors regarding the ground-truth. This result is essentially independent of the type of flow.
- There are no substantial differences between rotation-dominated, high vorticity flows, and deformation-dominated flows – the Lucas-Kanade/Liu-Shen and the Farnebäck/Liu-Shen combinations are the better performing methods in both cases. The accuracy of all methods depends mostly on stream-line curvature and velocity gradients. All methods exhibited difficulties in reproducing the inner region of the Poiseuille flow, where the velocity is zero at the wall and gradients are strong in the vicinity, and all flow singularities, where velocity gradients are strong around points of zero velocity.
- Taking PIV as a reference, the range of high performance of OpF methods is broader. While PIV has a well-known optimal range of particle number concentration of  $N_i \geq 6$  up to the speckle threshold, the Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen combinations exhibit high accuracy from  $N_i = 1$  up to  $N_i = 16$ , provided that the particle spot sizes are also within a optimal range. For correlation based-PIV, the optimal tracer spot diameter range is 2.0 px to 3.0 px (it can go up to 6.0 px in some PIV implementations). The Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen combinations consistently exhibit high accuracy between 2.0px and 6.0px tracer spot diameters.
- The Lucas-Kanade method and its variants work well with all tested image bit depths. Farnebäck is able to operate with 8-bit depth images but it is only competitive at 10-bit depths or higher. For all methods, the variation pattern of accuracy with bit-depth is intrinsic – it is not changed when employing pyramidal multi-resolution or combination with the Liu-Shen algorithm.
- Despite its generalized use, the basic differential method Horn-Schunck is the worst performing algorithm. Its 2-Level pyramidal version combined with Liu-Shen improves the results but always much less than the other methods subjected to the same treatment. Its lower accuracy and high parameter sensitivity are its main disadvantages. Horn-Schunck and their variants work best at an image bit depth of 8 bits, with slightly smaller performances for higher bit depths.

From this benchmarking exercise, we note that the OpF algorithms that seem more promising for further investment, in the scope of fluid mechanics applications, possibly with hybridization with PIV, are the dense Lucas-Kanade/Liu-Shen and Farnebäck/Liu-Shen combinations, the latter only for 10 bit or

higher images. This observation is independent of type of flow or imaging conditions. Evidently, users should make the effort to set up imaging conditions in the range that allows better performances.

A relevant application of this benchmarking exercise concerns the re-processing of existing PIV databases. Re-processing may allow for the extraction of denser velocity maps even if the imaging conditions are not optimal for PIV. This is consequence of the fact that the high performance range of OpF seems broader than that of the correlation-based PIV.

It is relevant to point out this results were obtained with synthetic images and simple planar flows of incompressible fluids. As future work, a validation should be carried out with real PIV images. In particular it is necessary to evaluate the consequences of the densification of the velocity maps in the presence of turbulent flows. It may also be interesting to consider solutions based on Artificial Intelligence (AI), possibly through deep learning, or more specifically Neural Networks (NN) or Convolutional Neural Networks (CNNs).

## 3.9 Acknowledgements

This research was partially supported by Portuguese and European funds, within the COMPETE 2020 and PORL-FEDER programs, through project PTDC/CTA-OHR/29360/2017 RiverCure and by the PhD grant SFRH/BD/137967/2018 from the Portuguese Foundation for Science and Technology (FCT).

## Conflict of interest

The authors declare that they have no conflict of interest.

## 3.A Appendices

### 3.A.1 PIV and optical flow methods

This sub-section is dedicated to bring PIV, in a loose sense, into context with OpF. We present in Figure 3.12 a brief comparison between JPIV v21.08 (Java), OpenPIV v0.23.6 (Python) and QuickLab PIV v0.5.7 (our reference implementation). From that figure we can see that all three PIV implementations perform better for particle concentrations of  $N_i \geq 6$  (higher reliability) as well as with particle spot sizes of 2.0 or 3.0 px diameters (higher precision) except OpenPIV which also deals well with 6.0 px particle spot sizes. Furthermore, none of the implementations is able to cope well with 1.0 px particle spot sizes. These PIV implementations match the theoretical expected behavior for PIV methods (see Raffel et al. 38).

OpF methods are compared with PIV by dividing all images in into interrogation areas of 16 px by 16 px, as employed by PIV software, rather than considering all the individual OpF vectors. Thus, a SIG absolute error and a SIG relative error are computed per each interrogation area based on the optical flow vectors that are contained in each interrogation area. Only at the final step, the average of the

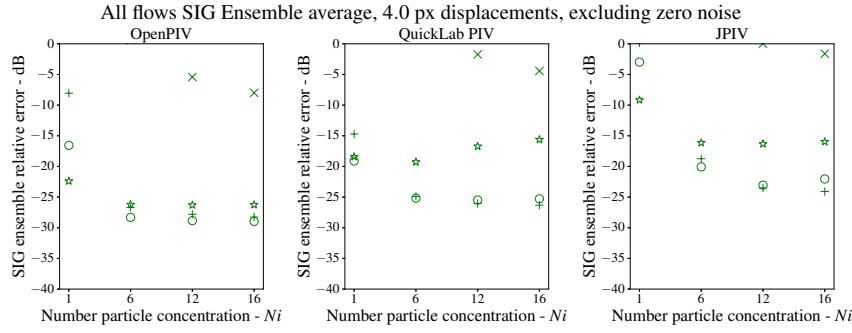


Figure 3.12: JPIV, OpenPIV and QuickLab PIV accuracy vs particle spot sizes and concentration ( $N_i$ ). Results shown are obtained by averaging the SIGs relative error across all flow types at 4.0 px maximum displacement. Zero noise data is excluded since it is not realistic.

errors for all the interrogation areas is taken mimicking the error computation process that is made for the PIV accuracy evaluation.

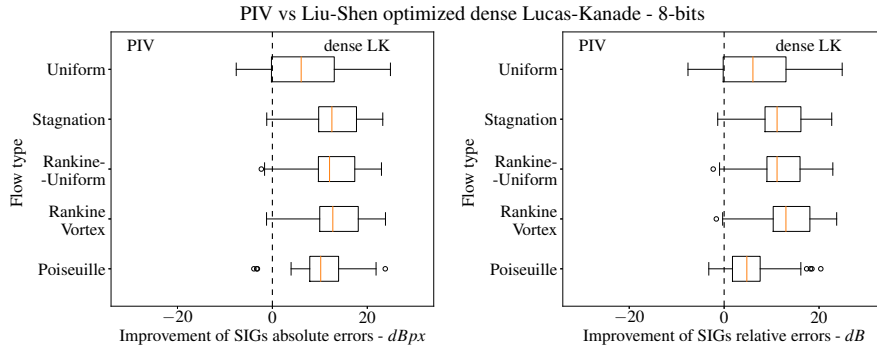


Figure 3.13: QuickLab PIV (PIV) versus dense Lucas-Kanade/Liu-Shen combined Pyramidal L2 box plots by flow type (8-bits) for the absolute ( $\epsilon - SIG$ ) and relative ( $\epsilon_R - SIG$ ) SIG errors

Observing Figure 3.13 we see that the combined dense Lucas-Kanade/Liu-Shen performs well when compared to our PIV reference implementation, both in terms of SIG absolute and relative error. A few combinations of parameters work marginally better with our PIV reference implementation, particularly with the uniform flow and displacements of 4.0 px ( $E = -0.16dBpx$  and  $E_R = -0.16dB$ ). Nevertheless, those exceptions do not present relevant expression and it also does not mean that the tested OpF method is superior to existing state of the art PIV solutions. Our PIV implementation is a classic PIV implementation and will likely not perform on par with the current state of the art solutions.

The comparison chart in Figure 3.13 confirms that the combined dense Lucas-Kanade/Liu-Shen method is valid for PIV processing of images with 8-bit depths. With an image bit depth of 10-bits it becomes possible to employ Farnebäck/Liu-Shen combination. The comparison results shown in Figure 3.14 confirm that this method also performs well in comparison to our PIV reference implementation, both in terms of SIG absolute and relative error. The only relevant exceptions, albeit only in occurrence number, not in error value, occur for the uniform flow and displacements of 4.0 px where  $E = -0.28dBpx$  and  $E_R = -0.28dBpx$ . From these results one can conclude that the Farnebäck/Liu-Shen combination is suitable for processing images with 10 or 12-bit depths. Our implementation of PIV does not show accuracy improvements with increasing bit depths above 8-bits.

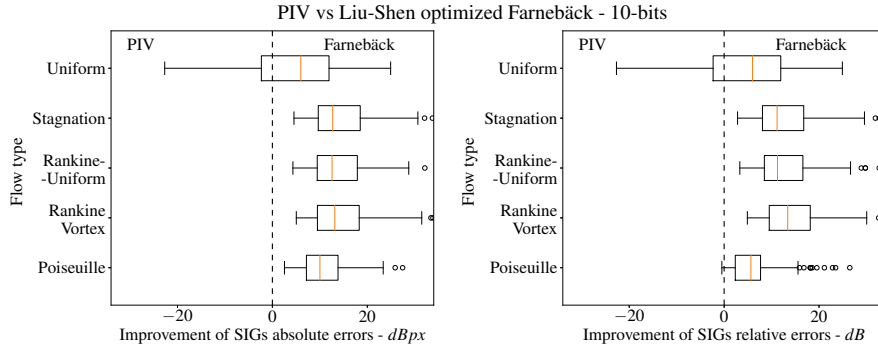


Figure 3.14: QuickLab PIV (PIV) versus Farneback/Liu-Shen combined Pyramidal L2 box plots by flow type (10-bits) for the absolute ( $\epsilon - SIG$ ) and relative ( $\epsilon R - SIG$ ) SIG errors

### 3.A.2 Pyramidal vs non-pyramidal optical flow and bit-depths

Table 3.5 details the effects of employing pyramidal processing with the base algorithms at the various image bit depth levels, as a function of the maximum velocity considered (0.05 IA, 0.10 IA, 0.25 IA). It forms the base for the discussion of the pyramidal effects on accuracy. The values obtained for this table are averages of the SIG errors (both absolute and relative) as described in section 3.4.

Table 3.5: Absolute and relative errors (Avg.) and standard deviations (Std.) of the base algorithms and of the base algorithms combined with 2-level pyramidal processing. All statistics are discriminated by displacement and are computed from SIG ensembles. Each SIG ensemble is composed of all flow types, particle spot sizes, particle concentrations and image bit depths.

| Max.<br>Velocity    | Algorithm name         | SIG Ensemble<br>Absolute Errors<br>(dBpx) |       | SIG Ensemble<br>Relative Errors<br>(dB) |       |
|---------------------|------------------------|---|-------|---|-------|
|                     |                        | Avg.                                      | Std.  | Avg.                                    | Std.  |
| 0.8 px<br>(0.05 IA) | Farneback              | -22.17                                    | 8.40  | -13.59                                  | 8.59  |
|                     | Farneback Pyramidal    | -22.16                                    | 8.41  | -13.58                                  | 8.61  |
|                     | Lucas-Kanade           | -22.58                                    | 11.52 | -14.03                                  | 11.38 |
|                     | Lucas-Kanade Pyramidal | -22.53                                    | 11.67 | -13.98                                  | 11.53 |
|                     | Horn-Schunck           | -15.82                                    | 6.05  | -8.00                                   | 5.87  |
|                     | Horn-Schunck Pyramidal | -15.87                                    | 6.11  | -8.02                                   | 5.97  |
| 1.6 px<br>(0.1 IA)  | Farneback              | -21.18                                    | 9.24  | -18.42                                  | 9.39  |
|                     | Farneback Pyramidal    | -21.17                                    | 9.26  | -18.42                                  | 9.41  |
|                     | Lucas-Kanade           | -21.89                                    | 12.05 | -18.87                                  | 12.28 |
|                     | Lucas-Kanade Pyramidal | -21.83                                    | 12.22 | -18.81                                  | 12.45 |
|                     | Horn-Schunck           | -12.64                                    | 8.10  | -10.99                                  | 7.29  |
|                     | Horn-Schunck Pyramidal | -12.72                                    | 8.08  | -11.05                                  | 7.32  |
| 4.0 px<br>(0.25 IA) | Farneback              | -18.95                                    | 10.08 | -24.06                                  | 10.13 |
|                     | Farneback Pyramidal    | -19.51                                    | 10.11 | -24.54                                  | 10.48 |
|                     | Lucas-Kanade           | -12.11                                    | 14.93 | -18.63                                  | 13.26 |
|                     | Lucas-Kanade Pyramidal | -21.38                                    | 10.87 | -25.85                                  | 12.40 |
|                     | Horn-Schunck           | -2.98                                     | 8.54  | -10.69                                  | 7.54  |
|                     | Horn-Schunck Pyramidal | -5.45                                     | 10.27 | -12.48                                  | 9.04  |

From Table 3.5 it can be seen that the base version of Farneback is similar to its two level pyramidal version, while Horn-Schunck shows a small improvement for larger displacements of 4.0 pixels in respect to the two level pyramidal version. Instead, Lucas-Kanade has significant benefits with the two level pyramidal variant for the larger displacements. The distribution of the errors shown in Figure 3.16 clarify

the overall benefits of the pyramidal implementation. When comparing Lucas-Kanade against its own two level pyramidal version, we find that the pyramidal version is able to track 4.0 pixel displacements (0.25 IA width) much better than the non-pyramidal versions (see Table 3.5 and Figure 3.15).

Table 3.6 discriminates the benefits across types of flow. It can be seen that dense Lucas-Kanade pyramidal version is especially better for the uniform flow, followed by the parabolic flow (see Table 3.6). Finally, rankine vortex, rankine with superimposed uniform and stagnation are only marginally improved. For smaller displacements, like 1.6 pixels (0.10 IA width) and 0.8 pixels (0.05 IA width), there is almost no measurable loss of accuracy, except a small error increment above the threshold for particles with 1.0 pixel diameter spots, with a particle number concentration per volume of 1 and a 15 dB WGIN (last entry of Table 3.6). Where, when considering a threshold of 1.0 dB for the  $E_R$  and 1.2 dB for the  $E$ , these thresholds are slightly exceeded.

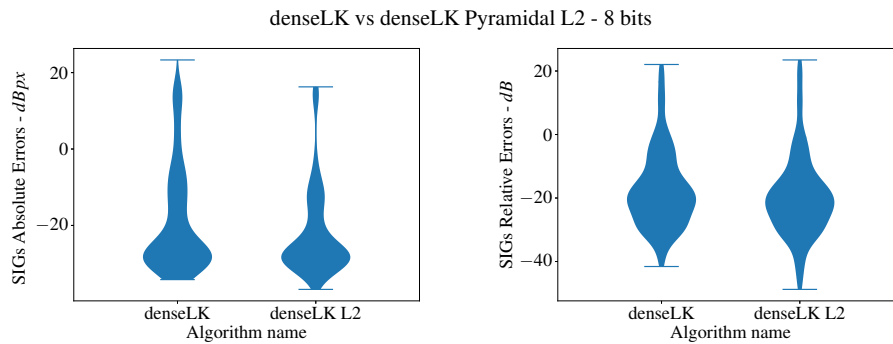


Figure 3.15: dense Lucas-Kanade versus dense Lucas-Kanade Pyramidal L2 violin plots for the absolute ( $\epsilon - SIG$ ) and relative ( $\epsilon_R - SIG$ ) SIG errors, all for 8-bit image depth.

Table 3.6: Pyramidal L2 dense Lucas-Kanade improvements over dense Lucas-Kanade per flow type-/displacement and bit depth. Where  $E$  is the SIG ensemble absolute error and  $E_R$  is the SIG ensemble relative error.

| Flow type<br>and max.<br>displacement                                   | 8 bits        |               | 10 bits       |               | 12 bits       |               |
|---|---------------|---------------|---------------|---------------|---------------|---------------|
|   | $E$<br>(dBpx) | $E_R$<br>(dB) | $E$<br>(dBpx) | $E_R$<br>(dB) | $E$<br>(dBpx) | $E_R$<br>(dB) |
| Uniform 4.0px   | 32.36         | 32.36         | 32.55         | 32.55         | 32.57         | 32.57         |
| Poiseuille 4.0px  | 16.57         | 7.33          | 16.66         | 7.39          | 16.66         | 7.41          |
| Rankine<br>vortex 4.0px   | 5.50          | 3.66          | 5.66          | 3.79          | 5.48          | 3.65          |
| Rankine w/<br>Uniform 4.0px   | 3.76          | 1.77          | 3.79          | 1.77          | 3.87          | 1.85          |
| Stagnation 4.0px  | 4.26          | 2.20          | 4.02          | 2.08          | 4.25          | 2.18          |
| $d_\tau = 1.0px$ ,<br>$N_i = 1$ ,<br>$WGIN = 15dB$ ,<br>0.8px and 1.6px | -1.30         | -1.33         | -1.36         | -1.41         | -1.43         | -1.35         |

Figure 3.15 shows that the pyramidal version provides accuracy improvements even at the lowest image bit-depth considered of 8-bits, where one can see that the errors are quite reduced. Moving to 10 or

12-bit image depths does not improve the accuracy for this method (see Table 3.6).

Table 3.7: Pyramidal L2 Farnebäck improvements over Farnebäck per flow type/displacement and bit depth. Where  $E$  is the SIG ensemble absolute error and  $E_R$  is the SIG ensemble relative error.

| Flow type and max. displacement | 8 bits          |               | 10 bits       |               | 12 bits       |               |
|---------------------------------|-----------------|---------------|---------------|---------------|---------------|---------------|
|                                 | $E$<br>(dBpx)   | $E_R$<br>(dB) | $E$<br>(dBpx) | $E_R$<br>(dB) | $E$<br>(dBpx) | $E_R$<br>(dB) |
| Uniform 4.0px                   | 1.93            | 1.93          | 4.57          | 4.57          | 5.69          | 5.69          |
| Poiseuille 4.0px                | $  < 1.2dBpx  $ | $  < 1.0dB  $ | 1.86          | 0.52          | 1.68          | 0.48          |

Referring again to Table 3.5, but now regarding to Farnebäck versus two level pyramidal Farnebäck, it can be seen that the accuracy benefits only occur with a 4.0 px maximum displacement. The Farnebäck non-pyramidal method is already capable of tracking 4.0 px displacements, better than the other methods, just not as good as the pyramidal version.

From Table 3.7, the Pyramidal results for Farnebäck can be seen in more detail. The benefits occur specifically at 4.0 px maximum displacement for the uniform flow for all the bit-depths, and for the Poiseuille flow for 10 and 12-bit depths. This is the only case where 1.2 dBpx SIG absolute error or the 1.0 dB SIG relative error are exceeded, resulting in slightly improved accuracy. There are no accuracy drawbacks for any of the displacements, 0.8 px, 1.6 px or 4.0 px which exceed the considered thresholds ( $> 1.2$  dBpx SIG absolute error or the  $> 1.0$  dB SIG relative error) for the tested bit depths. Figure 3.16 shows the accuracy improvement achieved for images with 12-bit image depths, being somewhat similar to what happens with 10-bit image depths, both with better results than the 8-bit image depths (see Table 3.7). For 8-bit image depths and 4.0 px displacements the gains are smaller, but the loss is not related with the pyramidal processing method, since the 8-bit depth non-pyramidal version also displays much worse performance than the higher bit-depths. As such, the accuracy loss lies in the base Farnebäck method, which does not perform so well for 8-bit depths.

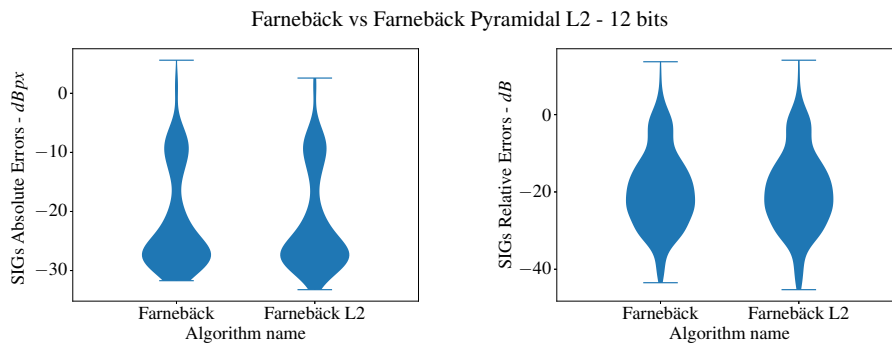


Figure 3.16: Farnebäck versus Farnebäck Pyramidal L2 violin plots for the absolute ( $\epsilon - SIG$ ) and relative ( $\epsilon R - SIG$ ) SIG errors, all for 12-bit image depth.

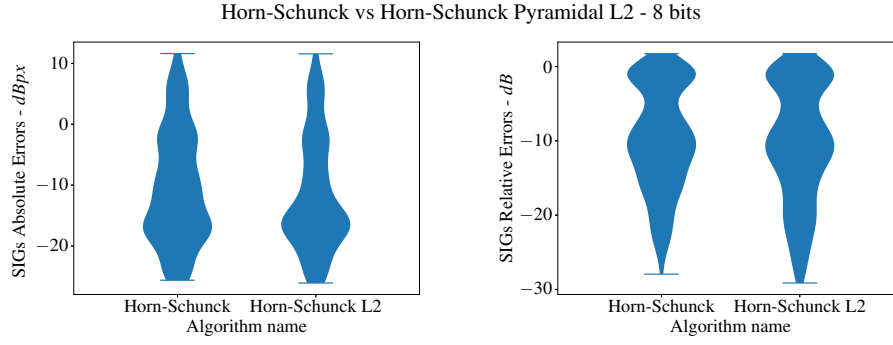


Figure 3.17: Horn-Schunck versus Horn-Schunck Pyramidal L2 violin plots for the absolute ( $\epsilon - SIG$ ) and relative ( $\epsilon_R - SIG$ ) SIG errors, all for 8-bit image depth.

Table 3.8: Pyramidal L2 Horn-Schunck improvements over Horn-Schunck per flow type/displacement and bit depth. Where  $E$  is the SIG ensemble absolute error and  $E_R$  is the SIG ensemble relative error.

| Flow type and max. displacement  | 8 bits        |               | 10 bits       |               | 12 bits       |               |
|--|---------------|---------------|---------------|---------------|---------------|---------------|
|  | $E$<br>(dBpx) | $E_R$<br>(dB) | $E$<br>(dBpx) | $E_R$<br>(dB) | $E$<br>(dBpx) | $E_R$<br>(dB) |
| Uniform 0.8px  | 1.09          | 1.09          | < 1.2dBpx     | < 1.0dB       | < 1.2dBpx     | < 1.0dB       |
| Uniform 1.6px  | 1.09          | 1.09          | < 1.2dBpx     | < 1.0dB       | < 1.2dBpx     | < 1.0dB       |
| Uniform 4.0px  | 12.07         | 12.07         | 11.55         | 11.55         | 10.64         | 10.64         |
| Poiseuille 4.0px   | 6.98          | 2.93          | 5.58          | 2.46          | 6.12          | 2.62          |
| Rankine vortex 4.0px   | 3.96          | 2.89          | 3.94          | 2.99          | 3.54          | 2.67          |
| Rankine w/ Uniform 4.0px   | 3.58          | 2.02          | 5.01          | 2.81          | 3.54          | 1.97          |
| Stagnation 4.0px   | 3.80          | 1.51          | 3.87          | 2.28          | 3.75          | 2.17          |
| Poiseuille,<br>$\tau = 3.0px$ ,<br>$N_i = 12$ ,<br>$WGIN = 0dB$ ,<br>1.6px | -1.36         | -0.70         | < 1.2dBpx     | < 1.0dB       | < 1.2dBpx     | < 1.0dB       |

When considering Horn-Schunck versus two level pyramidal Horn-Schunck, and referring to Table 3.5, one can observe a reduction in the average errors accompanied with an increase in the standard deviation. Still, despite spreading the errors across a larger range, relevant accuracy improvements are obtained for 4.0 px displacements (see Figure 3.17).

Table 3.8 further details the results for Horn-Schunck. From that table it can be seen that for 4.0 px maximum displacements the improvement occurs especially for the uniform flow, while also improving Poiseuille flow and slightly improving Rankine vortex, Rankine vortex with superimposed uniform flow and flow with stagnation point. The method has no accuracy reductions, for most displacements, except for one situation which comprehends a set of special conditions. It occurs for 8-bit depth and parabolic flow at 1.6 px maximum displacement and 3.0 px diameter particle at a particle volume concentration of



12 and 0 dB WGIN where a slight accuracy loss occurs exclusively for the SIG absolute error improvement that exceed the 1.2dB. The SIG relative error improvement, for the same parameters combination, is still within the 1.0 dB threshold, meaning that the error is not significant when compared to the absolute vector magnitude (see Table 3.8). Moving to 10 or 12-bit image depths does not improve the accuracy for this method.

Horn-Schunck's Lagrange multiplier parameter must be carefully selected for each pyramidal level, as it also significantly influences the attained results. In fact, it also significantly depends on the particle volume concentration and bit-depths.

Table 3.9: Absolute and relative errors (Avg.) and standard deviations (Std.) of the base algorithms combined with Liu-Shen without pyramidal processing versus with 2-level pyramidal processing. All statistics are discriminated by displacement and are computed from SIG ensembles. Each SIG ensemble is composed of all flow types, particle spot sizes, particle concentrations and image bit depths.

| Max.<br>Velocity    | Algorithm name         | SIG Ensemble<br>Absolute Errors<br>(dBpx) |       | SIG Ensemble<br>Relative Errors<br>(dB) |       |
|---------------------|------------------------|---|-------|---|-------|
|                     |                        | Avg.                                      | Std.  | Avg.                                    | Std.  |
| 0.8 px<br>(0.05 IA) | Farnebäck              | -25.21                                    | 8.93  | -16.55                                  | 9.02  |
|                     | Farnebäck Pyramidal    | -25.20                                    | 8.94  | -16.54                                  | 9.03  |
|                     | Lucas-Kanade           | -25.52                                    | 11.22 | -16.82                                  | 11.06 |
|                     | Lucas-Kanade Pyramidal | -25.50                                    | 11.29 | -16.80                                  | 11.14 |
|                     | Horn-Schunck           | -18.25                                    | 7.49  | -10.52                                  | 6.95  |
|                     | Horn-Schunck Pyramidal | -18.38                                    | 7.56  | -10.60                                  | 7.05  |
| 1.6 px<br>(0.1 IA)  | Farnebäck              | -24.17                                    | 9.92  | -21.23                                  | 9.98  |
|                     | Farnebäck Pyramidal    | -24.17                                    | 9.93  | -21.22                                  | 9.99  |
|                     | Lucas-Kanade           | -25.10                                    | 11.80 | -21.81                                  | 12.12 |
|                     | Lucas-Kanade Pyramidal | -25.07                                    | 11.89 | -21.78                                  | 12.21 |
|                     | Horn-Schunck           | -14.83                                    | 9.76  | -13.21                                  | 8.74  |
|                     | Horn-Schunck Pyramidal | -15.05                                    | 9.81  | -13.38                                  | 8.82  |
| 4.0 px<br>(0.25 IA) | Farnebäck              | -21.39                                    | 11.06 | -26.06                                  | 10.95 |
|                     | Farnebäck Pyramidal    | -22.14                                    | 10.87 | -26.82                                  | 11.19 |
|                     | Lucas-Kanade           | -14.61                                    | 14.97 | -21.08                                  | 13.27 |
|                     | Lucas-Kanade Pyramidal | -24.55                                    | 10.43 | -28.66                                  | 12.11 |
|                     | Horn-Schunck           | -3.92                                     | 9.47  | -11.76                                  | 8.56  |
|                     | Horn-Schunck Pyramidal | -7.29                                     | 12.13 | -14.27                                  | 10.80 |

When combining Liu-Shen optical flow with each of the considered base algorithms we can verify the same trend regarding non-pyramidal versus 2-level pyramidal. Details can be found in table 3.9. When employing 2-level pyramidal for the Liu-Shen / Lucas-Kanade there are accuracy improvements similar to the ones observed for the base Lucas-Kanade as shown in Figure 3.15. Regarding Liu-Shen / Farnebäck and its non-pyramidal versus 2-level pyramidal implementation, the behavior follows closely the one presented for the base Farnebäck like in Figure 3.16. Finally, the results for the Liu-Shen / Horn-Schunck pyramidal and non-pyramidal version are once again, quite similar when comparing with the base Horn-Schunck algorithm in Figure 3.17.

### 3.11 References

- [1] J. Carlier and B. Wieneke. Report 1 on production and diffusion of fluid mechanics images and data. fluid project deliverable 1.2. European Project'Fluid image analisys and description'(FLUID)—<http://www.fluid.irisa.fr>, 47, 2005. URL <http://fluid.irisa.fr/data-eng.htm>.
- [2] B. E. Schmidt and J. A. Sutton. High-resolution velocimetry from tracer particle fields using a wavelet-based optical flow method. Experiments in Fluids, 60(3):37, Feb 2019. ISSN 1432-1114. doi: 10.1007/s00348-019-2685-6. URL <https://doi.org/10.1007/s00348-019-2685-6>.
- [3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. International Journal of Computer Vision, 12(1):43–77, Feb 1994. ISSN 1573-1405. doi: 10.1007/BF01420984. URL <https://doi.org/10.1007/BF01420984>.
- [4] B. K. Horn and B. G. Schunck. Determining optical flow. Artificial Intelligence, 17(1):185–203, 1981. ISSN 0004-3702. doi: [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2). URL <https://www.sciencedirect.com/science/article/pii/0004370281900242>.
- [5] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'81, page 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. URL <https://dl.acm.org/doi/10.5555/1623264.1623280>.
- [6] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. SIAM Journal on Applied Mathematics, 60(1):156–182, 1999. doi: 10.1137/S0036139998340170. URL <https://doi.org/10.1137/S0036139998340170>.
- [7] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In J. Bigun and T. Gustavsson, editors, Image Analysis, pages 363–370, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45103-7. doi: 10.1007/3-540-45103-X\_50. URL [https://doi.org/10.1007/3-540-45103-X\\_50](https://doi.org/10.1007/3-540-45103-X_50).
- [8] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.
- [9] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision, 2(3):283–310, Jan 1989. ISSN 1573-1405. doi: 10.1007/BF00158167. URL <https://doi.org/10.1007/BF00158167>.
- [10] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. International Journal of Computer Vision, 68(2):125–143, Jun 2006. ISSN 1573-1405. doi: 10.1007/s11263-006-6660-3. URL <https://doi.org/10.1007/s11263-006-6660-3>.

- [11] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-7(2):139–154, 1985. doi: 10.1109/TPAMI.1985.4767639. URL <https://doi.org/10.1109/TPAMI.1985.4767639>.
- [12] P. Adam, B. Burg, and B. Zavidovique. Dynamic programming for region based pattern recognition. In ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 11, pages 2075–2078, 1986. doi: 10.1109/ICASSP.1986.1168952. URL <https://doi.org/10.1109/ICASSP.1986.1168952>.
- [13] D. J. Heeger. Optical flow using spatiotemporal filters. International Journal of Computer Vision, 1(4):279–302, Jan 1988. ISSN 1573-1405. doi: 10.1007/BF00133568. URL <https://doi.org/10.1007/BF00133568>.
- [14] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. International Journal of Computer Vision, 5(1):77–104, Aug 1990. ISSN 1573-1405. doi: 10.1007/BF00056772. URL <https://doi.org/10.1007/BF00056772>.
- [15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 2758–2766, 2015. doi: 10.1109/ICCV.2015.316. URL <https://doi.org/10.1109/ICCV.2015.316>.
- [16] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised deep learning for optical flow estimation. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1), Feb. 2017. URL <https://ojs.aaai.org/index.php/AAAI/article/view/10723>.
- [17] J. Sun, F. J. Quevedo, and E. Boltt. Bayesian optical flow with uncertainty quantification. Inverse Problems, 34(10):105008, aug 2018. doi: 10.1088/1361-6420/aad7cc. URL <https://doi.org/10.1088/1361-6420/aad7cc>.
- [18] G. Chantas, T. Gkamas, and C. Nikou. Variational-bayes optical flow. Journal of Mathematical Imaging and Vision, 50(3):199–213, Nov 2014. ISSN 1573-7683. doi: 10.1007/s10851-014-0494-3. URL <https://doi.org/10.1007/s10851-014-0494-3>.
- [19] S. Roth and M. J. Black. On the spatial statistics of optical flow. International Journal of Computer Vision, 74(1):33–50, Aug 2007. ISSN 1573-1405. doi: 10.1007/s11263-006-0016-x. URL <https://doi.org/10.1007/s11263-006-0016-x>.
- [20] I. Kajo, A. S. Malik, and N. Kamel. An evaluation of optical flow algorithms for crowd analytics in surveillance system. In 2016 6th International Conference on Intelligent and Advanced Systems (ICIAS), pages 1–6, 2016. doi: 10.1109/ICIAS.2016.7824064. URL <https://doi.org/10.1109/ICIAS.2016.7824064>.
- [21] J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. A Survey on Variational Optic Flow Methods for Small Displacements, pages 103–136. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN

- 978-3-540-34767-5. doi: 10.1007/978-3-540-34767-5\_5. URL [https://doi.org/10.1007/978-3-540-34767-5\\_5](https://doi.org/10.1007/978-3-540-34767-5_5).
- [22] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Velthuis, B. Li, and J. Yuan. A survey of variational and cnn-based optical flow techniques. Signal Processing: Image Communication, 72:9–24, 2019. ISSN 0923-5965. doi: <https://doi.org/10.1016/j.image.2018.12.002>. URL <https://www.sciencedirect.com/science/article/pii/S0923596518302479>.
- [23] D. Kondermann, S. Abraham, G. Brostow, W. Förstner, S. Gehrig, A. Imiya, B. Jähne, F. Klose, M. Magnor, H. Mayer, R. Mester, T. Pajdla, R. Reulke, and H. Zimmer. On performance analysis of optical flow algorithms. In F. Dellaert, J.-M. Frahm, M. Pollefeys, L. Leal-Taixé, and B. Rosenhahn, editors, Outdoor and Large-Scale Real-World Scene Analysis, pages 329–355, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-34091-8. doi: 10.1007/978-3-642-34091-8\_15. URL [https://doi.org/10.1007/978-3-642-34091-8\\_15](https://doi.org/10.1007/978-3-642-34091-8_15).
- [24] J. Westerweel. Digital particle image velocimetry: Theory and application. PhD thesis, Mechanical Maritime and Materials Engineering, Delft, June 1993.
- [38] M. Raffel, C. Willert, S. Wereley, and J. Kompenhans. Particle Image Velocimetry - A practical guide. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 978-3-642-43166-1. doi: 10.1007/978-3-540-72308-0. URL <https://doi.org/10.1007/978-3-540-72308-0>.
- [17] G. M. Quénot, J. Pakleza, and T. A. Kowalewski. Particle image velocimetry with optical flow. Experiments in Fluids, 25(3):177–189, Aug 1998. ISSN 1432-1114. doi: 10.1007/s003480050222. URL <https://doi.org/10.1007/s003480050222>.
- [18] P. Ruhnau, T. Kohlberger, C. Schnörr, and H. Nobach. Variational optical flow estimation for particle image velocimetry. Experiments in Fluids, 38(1):21–32, Jan 2005. ISSN 1432-1114. doi: 10.1007/s00348-004-0880-5. URL <https://doi.org/10.1007/s00348-004-0880-5>.
- [25] T. Liu and L. Shen. Fluid flow and optical flow. Journal of Fluid Mechanics, 614:253–291, 2008. doi: 10.1017/S0022112008003273. URL <https://doi.org/10.1017/S0022112008003273>.
- [29] B. Wang, Z. Cai, L. Shen, and T. Liu. An analysis of physics-based optical flow. Journal of Computational and Applied Mathematics, 276:62–80, 2015. ISSN 0377-0427. doi: <https://doi.org/10.1016/j.cam.2014.08.020>. URL <https://www.sciencedirect.com/science/article/pii/S037704271400377X>.
- [30] T. Liu, A. Merat, M. H. M. Makhmalbaf, C. Fajardo, and P. Merati. Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. Experiments in Fluids, 56(8):166, Aug 2015. ISSN 1432-1114. doi: 10.1007/s00348-015-2036-1. URL <https://doi.org/10.1007/s00348-015-2036-1>.
- [19] T. Corpetti, D. Heitz, G. Arroyo, E. Mémin, and A. Santa-Cruz. Fluid experimental flow estimation based on an optical-flow scheme. Experiments in Fluids, 40(1):80–97, Jan 2006. ISSN 1432-1114. doi: 10.1007/s00348-005-0048-y. URL <https://doi.org/10.1007/s00348-005-0048-y>.

- [20] P. Ruhnau and C. Schnörr. Optical stokes flow estimation: an imaging-based control approach. Experiments in Fluids, 42(1):61–78, Jan 2007. ISSN 1432-1114. doi: 10.1007/s00348-006-0220-z. URL <https://doi.org/10.1007/s00348-006-0220-z>.
- [21] D. Heitz, P. Héas, E. Mémin, and J. Carlier. Dynamic consistent correlation-variational approach for robust optical flow estimation. Experiments in Fluids, 45(4):595–608, Oct 2008. ISSN 1432-1114. doi: 10.1007/s00348-008-0567-4. URL <https://doi.org/10.1007/s00348-008-0567-4>.
- [22] D. Heitz, E. Mémin, and C. Schnörr. Variational fluid flow measurements from image sequences: synopsis and perspectives. Experiments in Fluids, 48(3):369–393, Mar 2010. ISSN 1432-1114. doi: 10.1007/s00348-009-0778-3. URL <https://doi.org/10.1007/s00348-009-0778-3>.
- [35] N. Papadakis and E. Mémin. A variational technique for time consistent tracking of curves and motion. Journal of Mathematical Imaging and Vision, 31(1):81–103, May 2008. ISSN 1573-7683. doi: 10.1007/s10851-008-0069-2. URL <https://doi.org/10.1007/s10851-008-0069-2>.
- [36] C. Cassisa, S. Simoens, V. Prinet, and L. Shao. Subgrid scale formulation of optical flow for the study of turbulent flow. Experiments in Fluids, 51(6):1739–1754, Dec 2011. ISSN 1432-1114. doi: 10.1007/s00348-011-1180-5. URL <https://doi.org/10.1007/s00348-011-1180-5>.
- [37] S. Cai, E. Mémin, P. Dérian, and C. Xu. Motion estimation under location uncertainty for turbulent fluid flows. Experiments in Fluids, 59(1):8, Nov 2017. ISSN 1432-1114. doi: 10.1007/s00348-017-2458-z. URL <https://doi.org/10.1007/s00348-017-2458-z>.
- [38] P. J. Burt. The Pyramid as a Structure for Efficient Computation, pages 6–35. Springer Berlin Heidelberg, Berlin, Heidelberg, 1984. ISBN 978-3-642-51590-3. doi: 10.1007/978-3-642-51590-3\_2. URL [https://doi.org/10.1007/978-3-642-51590-3\\_2](https://doi.org/10.1007/978-3-642-51590-3_2).
- [39] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. IEEE Transactions on Communications, 31(4):532–540, 1983. doi: 10.1109/TCOM.1983.1095851. URL <https://doi.org/10.1109/TCOM.1983.1095851>.
- [40] P. Dérian, P. Héas, C. Herzet, and E. Mémin. Wavelets and Optical Flow Motion Estimation. Numerical mathematics : a journal of Chinese universities, 6:116–137, 2013. URL <https://hal.inria.fr/hal-00737566>.
- [41] E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. International Journal of Computer Vision, 46(2):129–155, Feb 2002. ISSN 1573-1405. doi: 10.1023/A:1013539930159. URL <https://doi.org/10.1023/A:1013539930159>.
- [42] N. Papenberg, A. Bruhn, T. Brox, and J. Weickert. Numerical justification for multi-resolution optical flow computation. In International workshop on computer vision and image analysis (IWCVIA), volume 26, pages 7–12. IWCVIA03, 2003.
- [43] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. International Journal of Computer Vision, 67(2):141–158, Apr

2006. ISSN 1573-1405. doi: 10.1007/s11263-005-3960-y. URL <https://doi.org/10.1007/s11263-005-3960-y>.
- [24] F. Champagnat, A. Plyer, G. Le Besnerais, B. Leclaire, S. Davoust, and Y. Le Sant. Fast and accurate piv computation using highly parallel iterative correlation maximization. Experiments in Fluids, 50(4):1169, Mar 2011. ISSN 1432-1114. doi: 10.1007/s00348-011-1054-x. URL <https://doi.org/10.1007/s00348-011-1054-x>.
- [45] D. B. Bung and D. Valero. Optical flow estimation in aerated flows. Journal of Hydraulic Research, 54(5):575–580, 2016. doi: 10.1080/00221686.2016.1173600. URL <https://doi.org/10.1080/00221686.2016.1173600>.
- [23] G. Le Besnerais and F. Champagnat. Dense optical flow by iterative local window registration. In IEEE International Conference on Image Processing 2005, volume 1, pages I–137, 2005. doi: 10.1109/ICIP.2005.1529706. URL <https://doi.org/10.1109/ICIP.2005.1529706>.
- [47] O. Haggui., C. Tadonki., F. Sayadi., and B. Ouni. Efficient gpu implementation of lucas-kanade through openacc. In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, pages 768–775. INSTICC, SciTePress, 2019. ISBN 978-989-758-354-4. doi: 10.5220/0007272107680775. URL <https://doi.org/10.5220/0007272107680775>.
- [48] R. Allaoui, H. H. Mouane, Z. Asrih, S. Mars, I. El Hajjouji, and A. El mourabit. Fpga-based implementation of optical flow algorithm. In 2017 International Conference on Electrical and Information Technologies (ICEIT), pages 1–5, 2017. doi: 10.1109/EITech.2017.8255246. URL <https://doi.org/10.1109/EITech.2017.8255246>.
- [49] A. Plyer, G. Le Besnerais, and F. Champagnat. Massively parallel lucas kanade optical flow for real-time video processing applications. Journal of Real-Time Image Processing, 11(4):713–730, Apr 2016. ISSN 1861-8219. doi: 10.1007/s11554-014-0423-0. URL <https://doi.org/10.1007/s11554-014-0423-0>.
- [50] S. Lyasheva, R. Rakhmankulov, and M. Shleymovich. Frame interpolation in video stream using optical flow methods. Journal of Physics: Conference Series, 1488:012024, mar 2020. doi: 10.1088/1742-6596/1488/1/012024. URL <https://doi.org/10.1088/1742-6596/1488/1/012024>.
- [28] Z. Yang and M. Johnson. Hybrid particle image velocimetry with the combination of cross-correlation and optical flow method. Journal of Visualization, 20(3):625–638, Aug 2017. ISSN 1875-8975. doi: 10.1007/s12650-017-0417-7. URL <https://doi.org/10.1007/s12650-017-0417-7>.
- [29] G. Głomb and G. Świrniak. A hybrid method for velocity field of fluid flow estimation based on optical flow. In P. Lehmann, W. Osten, and A. A. G. Jr., editors, Optical Measurement Systems for Industrial

- Inspection XI, volume 11056, pages 969 – 980. International Society for Optics and Photonics, SPIE, 2019. doi: 10.1117/12.2525711. URL <https://doi.org/10.1117/12.2525711>.
- [53] J. H. Seong, M. S. Song, D. Nunez, A. Manera, and E. S. Kim. Velocity refinement of piv using global optical flow. Experiments in Fluids, 60(11):174, Oct 2019. ISSN 1432-1114. doi: 10.1007/s00348-019-2820-4. URL <https://doi.org/10.1007/s00348-019-2820-4>.
- [32] T. Liu, D. M. Salazar, H. Fagehi, H. Ghazwani, J. Montefort, and P. Merati. Hybrid optical-flow-cross-correlation method for particle image velocimetry. Journal of Fluids Engineering, 142(5), Feb 2020. ISSN 0098-2202. doi: 10.1115/1.4045572. URL <https://doi.org/10.1115/1.4045572>. 054501.
- [33] T. Liu and D. Salazar. Openopticalflow\_piv: An open source program integrating optical flow method with cross-correlation method for particle image velocimetry. Journal of Open Research Software, 9(1):3, 2021. doi: 10.5334/jors.326. URL <http://doi.org/10.5334/jors.326>.
- [56] K. Okamoto, S. Nishio, T. Saga, and T. Kobayashi. Standard images for particle-image velocimetry. Measurement Science and Technology, 11(6):685–691, may 2000. doi: 10.1088/0957-0233/11/6/311. URL <https://doi.org/10.1088/0957-0233/11/6/311>.
- [57] M. Stanislas, K. Okamoto, C. J. Kähler, and J. Westerweel. Main results of the second international piv challenge. Experiments in Fluids, 39(2):170–191, Aug 2005. ISSN 1432-1114. doi: 10.1007/s00348-005-0951-2. URL <https://doi.org/10.1007/s00348-005-0951-2>.
- [58] M. Stanislas, K. Okamoto, C. J. Kähler, J. Westerweel, and F. Scarano. Main results of the third international piv challenge. Experiments in Fluids, 45(1):27–71, Jul 2008. ISSN 1432-1114. doi: 10.1007/s00348-008-0462-z. URL <https://doi.org/10.1007/s00348-008-0462-z>.
- [59] C. J. Kähler, T. Astarita, P. P. Vlachos, J. Sakakibara, R. Hain, S. Discetti, R. La Foy, and C. Cierpka. Main results of the 4th international piv challenge. Experiments in Fluids, 57(6):97, May 2016. ISSN 1432-1114. doi: 10.1007/s00348-016-2173-1. URL <https://doi.org/10.1007/s00348-016-2173-1>.
- [60] B. Lecordier and J. Westerweel. The europiv synthetic image generator (s.i.g.). In M. Stanislas, J. Westerweel, and J. Kompenhans, editors, Particle Image Velocimetry: Recent Improvements, pages 145–161, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-642-18795-7. doi: 10.1007/978-3-642-18795-7\_11. URL [https://doi.org/10.1007/978-3-642-18795-7\\_11](https://doi.org/10.1007/978-3-642-18795-7_11).
- [61] L. Mendes, A. Bernardino, and R. M. Ferreira. piv-image-generator: An image generating software package for planar piv and optical flow benchmarking. SoftwareX, 12:100537, 2020. ISSN 2352-7110. doi: <https://doi.org/10.1016/j.softx.2020.100537>. URL <https://www.sciencedirect.com/science/article/pii/S2352711020300339>.

- [62] T. Liu. Openopticalflow: An open source program for extraction of velocity fields from flow visualization images. Journal of Open Research Software, 5(1):29, 2017. doi: 10.5334/jors.168. URL <http://doi.org/10.5334/jors.168>.
- [63] L. Mendes, A. Ricardo, and R. Ferreira. A customizable open-source piv software platform. In Proceedings of Hydrosenso 2019 - International Symposium and Exhibition on Hydro-Environment Sensors and Software, 2019.
- [64] G. Frost, B. Pitt, J. Freeman, L. Mendes, and et al. Aparapi v3.0.0 - a parallel api java open-source framework for gpgpu, 2021. URL <https://git.qoto.org/aparapi/aparapi/-/releases/v3.0.0>. Accessed 29 September 2021.
- [65] R. D. Keane and R. J. Adrian. Theory of cross-correlation analysis of piv images. Applied Scientific Research, 49(3):191–215, Jul 1992. ISSN 1573-1987. doi: 10.1007/BF00384623. URL <https://doi.org/10.1007/BF00384623>.
- [66] A. K. Prasad, R. J. Adrian, C. C. Landreth, and P. W. Offutt. Effect of resolution on the speed and accuracy of particle image velocimetry interrogation. Experiments in Fluids, 13(2):105–116, Jun 1992. ISSN 1432-1114. doi: 10.1007/BF00218156. URL <https://doi.org/10.1007/BF00218156>.
- [67] J. Westerweel. Fundamentals of digital particle image velocimetry. Measurement Science and Technology, 8(12):1379–1392, dec 1997. doi: 10.1088/0957-0233/8/12/002. URL <https://doi.org/10.1088/0957-0233/8/12/002>.



## Chapter 4

# QuickLabPIV new generation software

Article published in the MDPI Water journal with DOI: <https://doi.org/10.3390/w16071021>

Received: 19 February 2024, Revised: 20 March 2024, Accepted: 26 March 2024, Published: 1 April 2024.

The article was split into two chapters, this one being the first part, which was modified to just cover the Software and was augmented with a new section about the software design and computational performance. Otherwise it remains identical to the original published article.

### 4.1 Introduction

We present novel velocimetry algorithms based on the hybridization of correlation-based Particle Image Velocimetry (PIV) and a combination of Lucas–Kanade and Liu–Shen optical flow (OpF) methods. An efficient Aparapi/OpenCL implementation of those methods is also provided in the accompanying open-source QuickLabPIV-ng tool enabled with a Graphical User Interface (GUI). Two different options of hybridization were developed and tested: OpF as a last step, after correlation-based PIV, and OpF as a substitute for sub-pixel interpolation. Hybridization increases the spatial resolution of PIV, enabling the characterization of small turbulent scales and the computation of key turbulence parameters such as the rate of dissipation of turbulent kinetic energy.

Keywords: PIV densification   optical flow   hybrid PIV   OpF sub-pixel

### 4.2 Software Workflow and Main Features

#### 4.2.1 Workflow and Hybridization Options

We describe the main features and workflow of the hybrid PIV-OpF software, named QuickLabPIV-ng at version v0.8.7. The software is based on the PIV workflow. An initial coarse velocity vector field estimate is obtained by large, user-selected interrogation areas. This initial estimate is based on the cross-correlation of image pairs followed by correlation peak reconstitution to find the location of the

peak with sub-pixel accuracy. The initial vector field is then inherited by smaller interrogation areas that may be deformed for better matching. Between the PIV steps, OpF may be used as a substitute for peak reconstruction and peak location. At the end of all the PIV steps, OpF may be used as the last step either to fine-tune the PIV velocity estimate or to provide denser velocity maps. Again, the number of steps/cycles is predefined by the user when selecting the starting and ending IA sizes. Between PIV/OpF steps, a vector inspection may be carried out to validate the flow field. The substitution of wrong vectors may take place. The key aspects of the workflow are depicted in Figure 4.1. OpF methods can be either local or global. Local OpF methods operate exclusively with an image region and vectors in the vicinity of the vector under consideration. Global OpF methods depend on the overall image to estimate any single vector and thus cannot operate with small regions.

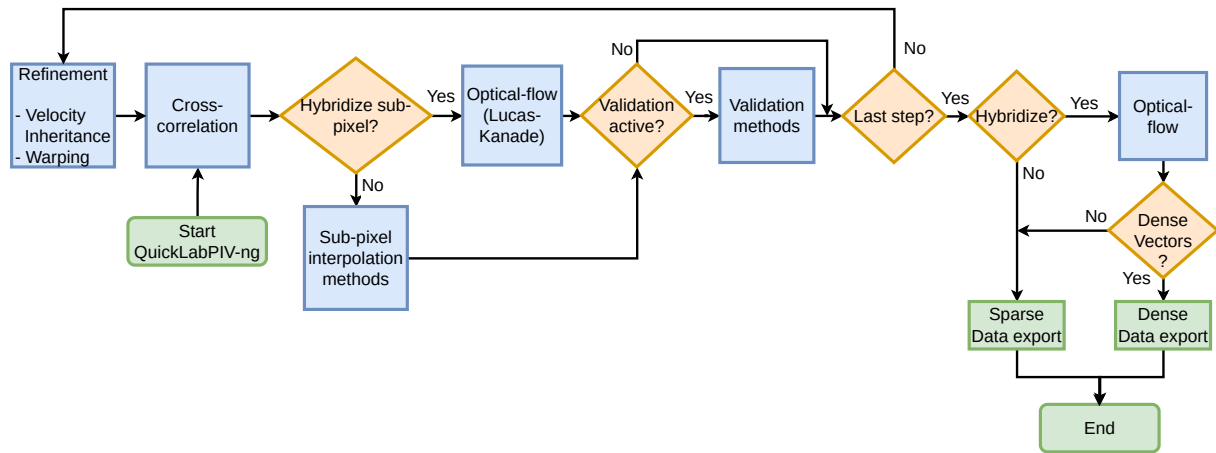


Figure 4.1: QuickLabPIV-ng—PIV processing steps with options for sub-pixel, validation, and OpF hybridization.

The software supports three different hybridization variants:

- Variant 1—“dense hybrid PIV-OpF”—employs dense optical flow as the last PIV step, either with dense Lucas–Kanade or with dense Liu–Shen combined with Lucas–Kanade. This option will provide densified velocity vector maps with 1 velocity vector per image pixel.
- Variant 2—“sub-pixel OpF”—employs optical flow as a substitution for the cross-correlation peak reconstruction and sub-pixel interpolation to find the coordinates of the peak. This is only valid for local optical flow methods like Lucas–Kanade. It directly extracts the sub-pixel movement from the original image by considering the pixels in the vicinity of the center of each IA, thus constituting an alternative to the PIV sub-pixel.
- Variant 3—“sparse hybrid PIV-OpF”—is similar to variant 1 but keeps only the velocity vectors of the center of each IA. We refer to this as sparse optical flow (1 velocity vector per each IA), thus keeping the original PIV vector resolution but allowing sub-pixel resolution. This supports both local and global optical flow methods, where a half-pixel warp is performed to obtain velocity pixels aligned with the center of each IA.

The process is iterated until the end of the last adaptive PIV step. Finally, the data are exported in

MATLAB file format.

The following sections describe all relevant PIV and OpF methods and algorithms.

#### **4.2.2 PIV Key Steps—Cross-Correlation and Sub-Pixel Interpolation**

QuickLabPIV-ng features IA cross-correlation via a 2D radix-2 FFT as the main PIV method, employing adaptive multi-step processing with customizable IA window sizes. It offers various sub-pixel interpolation methods, including traditional 1D-1D Gaussian three-point polynomial interpolation [1] and 1D-1D Gaussian robust LR [2]. Additionally, it supports optical flow methods like Lucas–Kanade and its combination with Liu–Shen; see Section 4.2.3.

#### **4.2.3 OpF Methods**

The Lucas–Kanade and Liu–Shen optical flow (OpF) methods enhance fluid mechanics analysis by adapting to a wide range of imaging conditions and particle dynamics (bit depths, particle concentration, spot sizes) [3]. Lucas–Kanade, a local method, calculates displacement within a defined window, iteratively refining estimates (more details in [4]). Conversely, Liu–Shen, a global approach, improves the initial velocity estimates by incorporating physical and optical effects of image formation (e.g., light scattering and intensity), requiring a robust starting point from other methods (more details in [5]). Both techniques necessitate pre-filtering for accuracy and support adaptive iteration counts. While Lucas–Kanade quickly converges, Liu–Shen demands more iterations for refinement. However, Liu–Shen is capable of improving the accuracy over the initial velocity estimations obtained from several other methods [3]. Additionally, the software accommodates both dense and sparse applications, offering a specialized Liu–Shen mode for clipped images, allowing CPU implementation, albeit with compromised precision.

#### **4.2.4 Hybridization of OpF and PIV after Correlation Steps**

QuickLabPIV-ng enhances PIV analysis by enabling the use of two distinct sub-pixel interpolation methods at various stages, optimizing the initial and final interrogation steps with separate techniques. This dual approach allows for the initial use of either sparse OpF or traditional sub-pixel methods, transitioning to dense OpF methods for final refinement. This strategy not only improves sub-pixel accuracy, particularly in the last steps, but also densifies velocity maps for more detailed results. The software’s flexibility in combining OpF and traditional methods offers significant advantages in precision and computational efficiency, bypassing traditional warping methods for a direct, accurate sub-pixel resolution. While standard sub-pixel techniques determine displacement by analyzing the geometry of cross-correlation peaks, optical flow (OpF) methods directly compute fractional displacement from the raw image. When the interrogation area (IA) size in the final PIV step is smaller than the OpF window, using dense OpF becomes more cost-effective and reduces the drawbacks of global methods. An OpF approach can bypass traditional PIV warping, offering a direct transition to precise sub-pixel measurements, maintaining an accuracy comparable to that of conventional warping techniques.

### 4.2.5 Vector Inheritance and Warping

When iterating to the next PIV step, i.e., smaller IA sizes, the processing continues with the refinement of interrogation areas (IAs) into smaller areas and includes a vector inheritance strategy, as well as an IA image warping strategy. When all adaptive PIV steps are completed, PIV processing proceeds with a possible hybridization (either variant 1 or variant 3). If hybridization is to be performed, then a single pass of optical flow is applied. We refer to methods featuring image warping as “Modern PIV”.

#### Vector Inheritance

The vector inheritance strategy involves selecting a method—area, distance, or bi-cubic spline interpolation—to import velocity vectors before adjusting the second image’s interrogation area (IA) windows accordingly for cross-correlation. Each method offers a unique way to calculate new IA displacements based on previous steps: area inheritance uses a weighted average from overlapping IA areas (see Figure 4.1), distance inheritance applies weights inversely proportional to the center distances of adjacent IAs, and bi-cubic spline inheritance interpolates displacements using a 2D map of IA centers. The Bi-cubic spline is noted for its superior accuracy in validations with synthetic image databases (see Section 5.2).

#### IA Image Warping

Window warping PIV employs backward image warping or a blend of window displacement and image warping, rather than shifting interrogation areas (IAs) by inherited displacements. This approach includes full or partial warping methods—traditional warping, mini-warping, and micro-warping. Traditional warping applies bi-cubic spline-interpolated displacements to each pixel, ensuring no voids or overlaps and maintaining fixed IA window locations. Mini-warping and micro-warping adjust IA windows based on estimated displacements, with micro-warping focusing on integer displacements followed by fractional warping for precision. QuickLabPIV-ng supports various configurations for applying warping to enhance PIV accuracy, with micro-warping on the second image showing the best results in validations; see Section 5.2.

### 4.2.6 Vector Validation and Substitution

Vector validation in PIV systems identifies and addresses incorrect velocity vectors caused by a low SNR, insufficient correlatable data, or other issues like boundary effects and particle movement. QuickLabPIV-ng offers methods to mark invalid vectors as zero or “not a number” for data export and to correct for such vectors where feasible, using a combination of validation and vector replacement techniques. These techniques ensure the reliability of the velocity data by effectively identifying and addressing outliers.

QuickLabPIV-ng implements two primary methods for vector validation: difference validation and normalized median validation [1]. The first evaluates the differences in the velocity vector within a local grid,

flagging those that exceed a set threshold as invalid. The latter uses a median-based approach, comparing each vector with the median vector within its vicinity against a normalized threshold to determine validity.

$$\frac{\|u_{med} - u(i, j)\|}{r_{med} + \epsilon_0} \leq \theta_{threshold}, \quad (4.1)$$

where  $u_{med}$  denotes the vector with the median Euclidean norm of all the vectors in the neighbor IAs;  $u(i, j)$  is the vector under analysis;  $r_{med}$  is the median of the residuals  $r_i = \|u_i - u_{med}\|$ , where  $i = 1, \dots, 8$  indexes the neighbor IAs; and  $\epsilon_0$  is a regularization term between 0.1 and 0.2 pixels.

QuickLabPIV-ng offers two methods for vector replacement: bi-linear and multi-peak. Bi-linear interpolation generates a new vector from valid neighboring vectors, while multi-peak searches for alternative high-correlation peaks as potential replacements, assuming the primary peak might be noise-distorted. If multi-peak fails, it defaults to bi-linear interpolation. The software also supports iterative validation to refine vector accuracy, though with caution to avoid misjudging valid vectors as invalid. This process enhances data reliability, especially in low-signal-to-noise scenarios.

## 4.3 Software and Graphical User Interface design

### 4.3.1 Software design

A newly released QuickLabPIV-ng (new generation) open-source software that supports the PIV hybrid methods of Liu-Shen combined with Lucas-Kanade and dense Lucas-Kanade method is made freely available for non-commercial use at <https://github.com/CoreRasurae/QuickLabPIVng>. The software is designed to be self-contained in terms of key numerical computations, supporting the PIV and Optical Flow methods, as this avoids possible behavioral changes that may modify the results due to framework/libraries implementation changes. This includes image filtering, image warping, cross-correlation computation, and Lucas-Kanade and Liu-Shen optical flow methods implementations. Most other PIV open-source software packages do not follow this approach, being dependent on several external libraries/frameworks for key PIV computations. As a side effect, our approach also makes all key computations easy to validate and audit, since only the required code is present and not a generic implementation for multiple final applications as would be in a library/framework. The software is also inherently CPU parallel and GpGPU massively parallel, to maximize throughput. The software was conceived in the Java language, being one of the most popular and well established programming languages, worldwide, that allows the same code to run on multiple computing platforms without having to be recompiled and without incurring significant performance penalties. The software was designed with design patterns (see Gamma, Helm, Johnson and Vlissides (1995)) and object oriented concepts in mind, resulting in good encapsulation and loosely coupled code with reduced inter-dependencies. The GUI presentation layer is separated from the business logic, so that alternate presentation layers can be easily integrated, including remote operation. The business logic can also be reused in another application as a hybrid PIV framework. QuickLabPIV-ng leverages the Aparapi (see Frost, Pitt, Freeman, Mendes and et al. (2021)) Java framework to massively parallelize, whenever possible, both the cross-

correlation, as well as the optical flow algorithms in CPU/GPU through OpenCL (see Khronos (2020)). The computation work can also be split across multiple OpenCL devices, assigning a Java thread to a given OpenCL device. When no OpenCL device is available, QuickLabPIV-ng is still able to parallelize work through multi-threading based on pure Java alternate implementations.

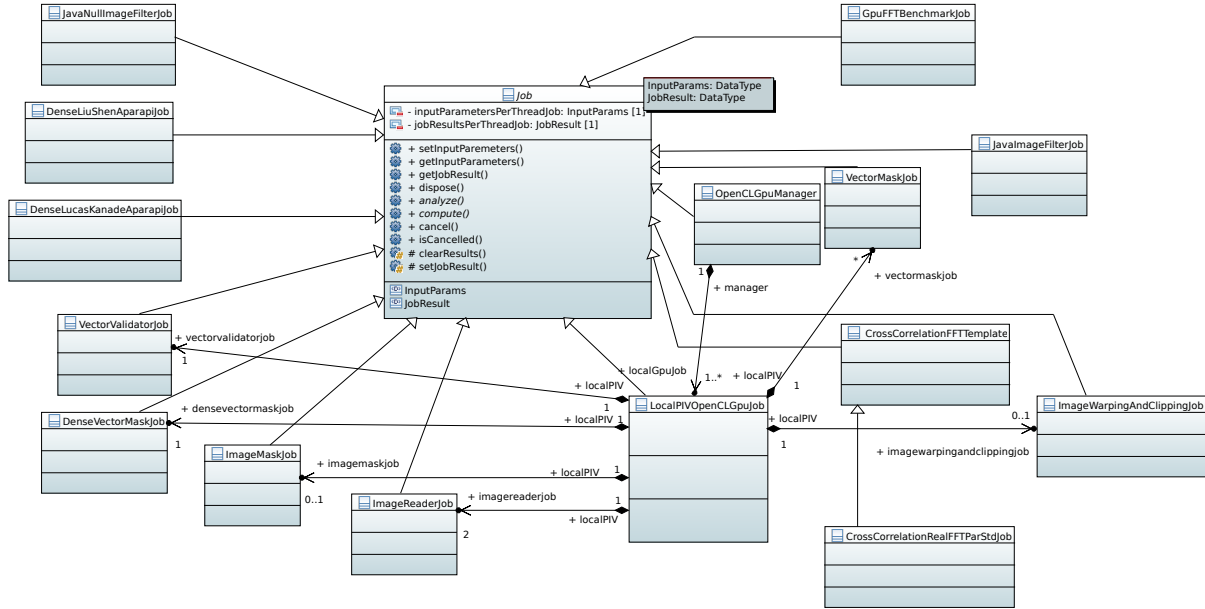


Figure 4.2: Unified Modeling Language (UML) simplified class diagram depicting the Job class software design.

The software design is centered around the Job concept as depicted in Figure 4.2. The idea is that the Job class abstracts operations that may run on different devices, that the operations may need to reserve or free resources, that they may require data transfers to another device and that data may need to be prepared, evaluated and re-organized before processing. By defining the templated Job class we define a common interface for all those operations, in a generalized way, that is able to accommodate multiple Job types, which can be sub-grouped together by type. All Jobs that belong to the same sub-group will share the same template data types for data input and result output, thus allowing an easy introduction of new PIV/OpF pre-processing, processing, or post-processing pipelines, while also helping with possible future custom software extensions via software plugins. Concrete Job specializations will then define the actual behavior, working as Adapters [6]) or Facades [6]) to the underlying support frameworks/APIs either internally or externally to QuickLabPIV-ng. If we look more closely to Figure 4.2 we see that the PIV vector validation, the image filtering, the dense OpF Lucas-Kanade, the dense OpF Liu-Shen and the FFT based cross correlation operations are all Jobs and that some job operations run on OpenCL GPU or CPU devices, while others run in CPU through the Java Virtual Machine (JVM). We also see that there are some Jobs that are compositions with other Jobs, as is the case, for instance with the LocalPIVOpenCLGpuJob that defines an Hybrid PIV/PIV pipeline and runs locally on a single machine. That some of its operations/jobs are likely accelerated by OpenCL devices and that this specific PIV processing pipeline manages the creation of the other required Jobs that it depends on for carrying out the PIV processing, e.g., like reading images, filtering images, computing

the cross-correlation, computing OpF, etc. A natural future extension for the LocalPIVOpenCLGpuJob is the DistributedPIVOpenCLGpuJob that would be a composition with several LocalPIVOpenCLGpuJob instances each running in their own machine in a distributed parallel manner, where each machine processes part of the PIV image database. The inputs and outputs of both Jobs would be the same, since they both perform the same task, that is, they both compute Hybrid PIV or regular PIV. The only difference is that they compute the data differently, however that is abstracted by the Job itself and the user can seamlessly change one processing method by the other. To help with these QuickLabPIV-ng also leverages the Concrete Factory design pattern, where the concrete Job instance is dynamically created at runtime according to the user processing parameterizations, which can be accessed throughout all the application via the Singleton design pattern [6]). Let us now look in more detail to the methods signature in the Job class. The clients of a given job group/type call:

- setInputParameters() to provide the generic input data.
- then at some point they call the analyze() method – responsible for evaluating and re-organizing the input data in a manner that is best suited to the processing method, and throwing an exception on any invalid, incorrect, or unsuitable data input for the given method.
- at a later stage with the data validated, re-organized and prepared for processing, the compute() method can be called to proceed with the computation on the requested device. Note that this may involve data transfers back and forth to a given device that is going to do the actual processing, e.g., a GPU, a FPGA.
- at any point during processing, the cancel() method can be called to proceed with the job cancellation request.
- isCancelled() can be called at any time to check if the job has been effectively cancelled.
- if the job was computed and no cancellation was requested, the results are available in a common format to all jobs of the same type, through the method getJobResult().
- dispose() is available for freeing memory or device resources that may have been allocated for the Job, including object pooling.

The actual class specialization calls clearResults() to remove the results of previous computation and calls setJobResult() for setting the results at the end of the Job computation. The QuickLabPIV-ng application is split into three main internal packages, namely, QuickLabPIV-ng, QuickLabPIV-ng\_UI and QuickLabPIV-lib as illustrated in Figure 4.3. QuickLabPIV-ng main external application dependencies are on the Aparapi, Log4j, JAXB, Apache Commons Math and TwelveMonkeys ImageIO frameworks through their respective Java JAR files (some of them not represented in the component diagram). Internally QuickLabPIV-ng application is packaged with a dependency on QuickLabPIV-lib, which provides some utility classes obtained on the Internet for managing Java Swing (Graphical User Interface) GUI components. The application is additionally split into QuickLabPIV-ng and QuickLabPIV-ng\_UI, where the first provides the business logic, that is, the hybrid PIV and regular PIV processing APIs, including

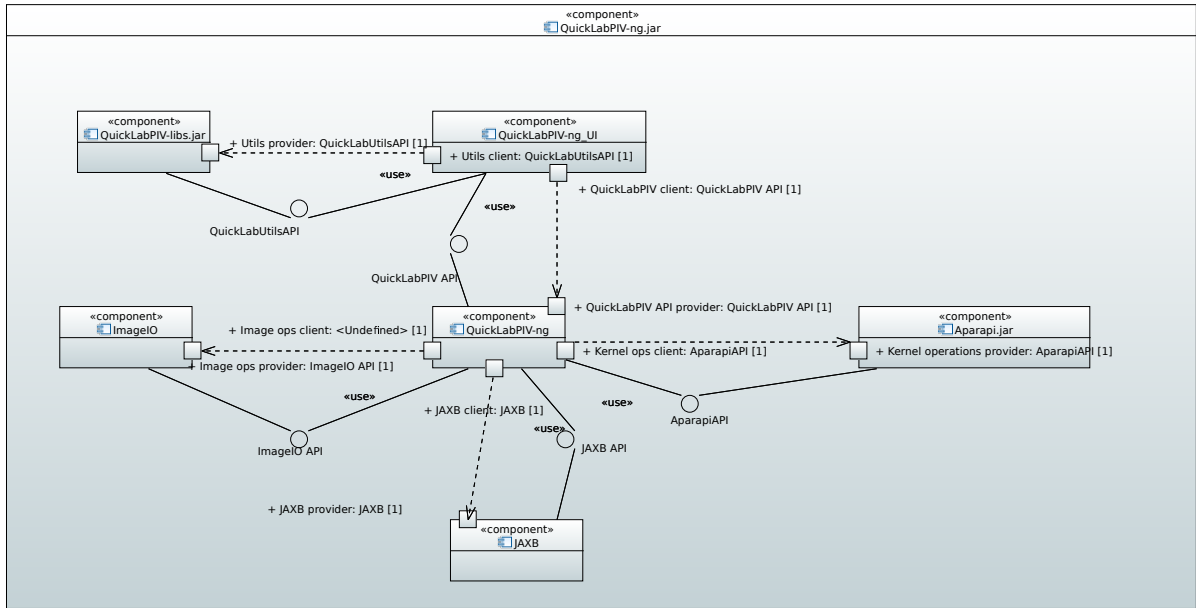


Figure 4.3: Unified Modeling Language (UML) simplified component diagram describing the main software components.

image loading and MATLAB® data file format export, whereas the latter provides the Swing GUI dialog windows. It should be noted that QuickLabPIV-ng and QuickLabPIV-ng\_UI are decoupled through the business logic API and that QuickLabPIV-ng\_UI employs the Model-View-Controller (MVC) design pattern and that navigation between dialog windows is carried by navigation controller objects. This design makes it possible to either use QuickLabPIV-ng as a hybrid PIV framework that can be included in larger applications, or directly, as an end user application with GUI. All QuickLabPIV-ng dependencies are packaged under a fat JAR file, named QuickLabPIV-ng.jar.

### 4.3.2 Graphical User Interface

All relevant QuickLabPIV-ng configurations can be easily done from a Graphical User Interface (GUI), despite not having all the desired usability characteristics, it already facilitates the end user when setting the execution and project configurations by comparison to a command-line application. The GUI was designed to be separate from the main application so that it can be easily adapted for remote operation, or by command-line, without the need to restructure the application. A sample GUI configuration window screenshot taken from the application is shown in Figure 4.4.

A PIV/Hybrid configuration along with the execution environment is stored in a project file in a local folder selected by the user and can be retrieved later through the application menu with the "Load Project" menu option. The GUI supports selecting all the PIV and Hybrid PIV configurations and includes configuration windows for the Execution Environment, both selecting the CPU cores and their assignment to selected GPUs, if available, and also allows to select the export options.



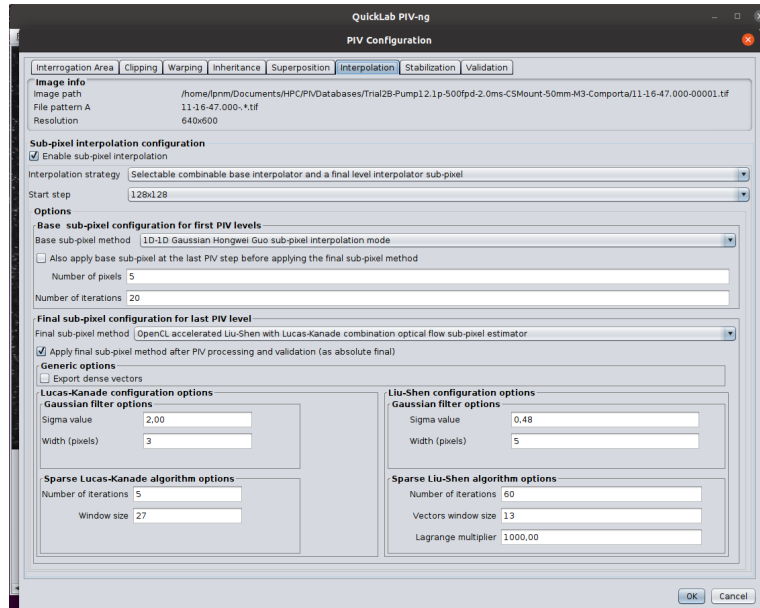


Figure 4.4: QuickLabPIV-ng GUI window for the PIV/Hybrid PIV sub-pixel interpolation configuration.

## 4.4 Computational Performance

In terms of QuickLabPIV-ng computation time-wise performance regarding PIV and Hybrid PIV processing, we have considered two computational platforms - Platform1 and Platform2 - over two sets of 500 PIV images pairs. One image set from a database with a resolution of  $1200 \times 1600$  pixels<sup>2</sup>, named Database1 and another set from a database of  $1920 \times 1200$  pixels<sup>2</sup>, named Database2. Both Windows and Linux operating systems were considered for the benchmark, but only Platform2 was benchmarked for both systems, while Platform1 is Linux only.

**Platform1:** Laptop test system for this performance benchmark, with the following characteristics:

- Lenovo ideapad GAMING 3 (15ACH6)
- AMD Ryzen 7 CPU 5800H 8-cores, 16-threads CPU
- 32 GB DDR4 RAM with 4 GB reserved for the AMD Renoir integrated GPU
- NVIDIA GeForce RTX 3050Ti with 4 GB VRAM discrete GPU
- Western Digital Blue NVMe SN570 1TB boot/operating system drive with EXT4 LUKS protected filesystem
- Ubuntu Desktop 20.04.6 LTS
- NVIDIA Driver 575.171.04
- OpenJDK 11.0.22 64-bit VM
- QuickLabPIV-ng v0.8.7 with 5 Java CPU threads and 1 OpenCL NVIDIA(R) RTX 3050Ti GPU and 27 GB maximum Java JVM allocation pool memory.

**Platform2:** Workstation test system for this performance benchmark, with the following characteristics:

- ASRock RAck EP2C602 server motherboard
- 2× Intel Xeon CPU E5-2650 v2 @ 2.60GHz supporting 8-cores/16-threads each
- 64 GB DDR3-1333 ECC Registered RAM
- AMD Radeon RX 5700 GPU with 8 GB of dedicated VRAM
- Samsung SATA SSD 860 EVO 1TB boot/operating system drive with EXT4 LUKS protected filesystem
- Dual-boot system with Ubuntu Desktop 20.04.6 LTS and Microsoft Windows 10 64-bits
- Linux AMDGPU driver version 6.0.60002/OpenCL 3602.0(HSA 1.1,LC)
- OpenJDK 11.0.22 64-bit VM
- QuickLabPIV-ng v0.8.7 with 6 Java CPU threads and 1 OpenCL AMD RX5700 GPU and 32 GB maximum Java JVM allocation pool memory.

**QuickLabPIV-ng Configurations:** Several PIV and hybrid PIV configurations are considered across all systems and platforms, namely:

- **Hybrid config. 1** - Hybrid PIV with Liu-Shen combined with dense Lucas-Kanade / dense export  
- Combined sub-pixel method with Hongwei Guo 3 pixel wide, 20 iterations sub-pixel for PIV steps, and a final step with Liu-Shen combined with dense Lucas-Kanade after PIV, followed by dense velocity field export. Both PIV and OpF accelerated by the GPU with Aparapi/OpenCL.
- **Hybrid config. 2** - Hybrid PIV with Liu-Shen combined with dense Lucas-Kanade / sparse export  
- Combined sub-pixel method with Hongwei Guo 3 pixel wide, 20 iterations sub-pixel for PIV steps, and a final step with Liu-Shen combined with dense Lucas-Kanade after PIV, followed by sparse velocity field export. Both PIV and OpF accelerated by the GPU with Aparapi/OpenCL.
- **Hybrid config. 3** - Hybrid PIV with dense Lucas-Kanade / dense export - Combined sub-pixel method with Hongwei Guo 3 pixel wide, 20 iterations sub-pixel for PIV steps, and a final step with dense Lucas-Kanade after PIV, followed by dense velocity field export. Both PIV and OpF accelerated by the GPU with Aparapi/OpenCL.
- **Hybrid config. 4** - Hybrid PIV with dense Lucas-Kanade / sparse export- Combined sub-pixel method with Hongwei Guo 3 pixel wide, 20 iterations sub-pixel for PIV steps, and a final step with dense Lucas-Kanade after PIV, followed by sparse velocity field export. Both PIV and OpF accelerated by the GPU with Aparapi/OpenCL.
- **PIV config. 1** - (OpenCL accelerated) - Hongwei Guo 3 pixel wide, 20 iterations, using OpenCL acceleration on a GPU device

- **PIV config. 2** - PIV (Java CPU only - 1 CPU thread) - Hongwei Guo 3 pixel wide, 20 iterations, using CPU only on a single core, no GPU acceleration.)
- **PIV config. 3** - PIV (Java CPU only - 8, or 16 CPU threads) - Hongwei Guo 3 pixel wide, 20 iterations, using CPU with all 8 cores in Platform1, or all 16 cores in Platform2, no GPU acceleration.
- **PIV config. 4** - PIV with the first PIV adaptive steps using Hongwei Guo 3 pixel wide, 20 iterations sub-pixel method and the last PIV adaptive step employing Lucas-Kanade sub-pixel. In this method both the Lucas-Kanade sub-pixel and the PIV cross-correlation are OpenCL accelerated.
- **PIV config. 5** - PIV with the first PIV adaptive steps using Hongwei Guo 3 pixel wide, 20 iterations sub-pixel method and the last PIV adaptive step employing Lucas-Kanade sub-pixel. In this method both the Lucas-Kanade sub-pixel and the PIV cross-correlation are Java CPU only, that is, non OpenCL accelerated.

All computations employ the same base PIV configuration with  $128 \times 128$  pixel<sup>2</sup> down to  $16 \times 16$  pixel<sup>2</sup> adaptive IAs size, 50% overlap, sub-pixel applied across all the adaptive steps. The PIV processing also employs micro-warping on the second image and PIV validation was applied with a normalized median validation mode with multi-peak substitution - 4 peaks with a 3 pixel kernel size - and the normalized median employs a threshold factor of 2.00 and  $\epsilon_0$  set to 0.15.

The processing times for Platform1 - Lenovo ideapad 3 laptop with an AMD Ryzen 7 5800H and a NVIDIA GeForce RTX 3050Ti 4GB VRAM - are shown in Table 4.1, while Table 4.2 and Table 4.3 show the processing times for Platform2, an Intel Xeon dual-CPU workstation with 16 cores total, 64GB ECC RAM and an AMD RX 5700 GPU. It should be noted that the QuickLabPIV-ng application lays out some support for optimized GpGPU OpenCL processing, however it has not been effectively tuned for all the GPU specific parameters. GPUs have very different architectures between brands and even between models of the same brand, making it difficult to have a general tuning rule, with many computing parameters to take into account. Some of these parameters include: number of compute units; ideal warp size; private, local and global memory sizes; data type memory alignment; number of threads per compute unit and maximum work group size. Although QuickLabPIV-ng has some mechanisms to adapt to each GPU specifics, namely it has support for extra details about each GPU model, including number of cores, preferred thread sizes, and existing algorithms to adapt the load configuration to each GPU, there was not enough time to tune and perfect these mechanisms in the software. Also, Aparapi is working reliably, but is not fully thread optimized, meaning that it compiles the kernels many more times than strictly needed and includes thread locking overheads. It should be noted that the PIV sub-pixel with sparse Lucas-Kanade OpF that is available in the current QuickLabPIV-ng (v0.8.7) version is computed entirely by a single Java thread. It is not parallelized in any way, i.e., the code is not vectorized, there is no OpenCL implementation for the sparse Lucas-Kanade and it is not multi-threaded. By contrast the dense Lucas-Kanade employed in hybrid PIV explores vectorization through Single Instruction Multiple Thread (SIMT) with an OpenCL implementation for GpGPU, thus running faster.

It should be noted that the execution times from Platform1 are not directly comparable to those of Platform2, since Platform1 ran with less 1 CPU thread and a lower JVM pool memory limit, 27 GB against

Table 4.1: Platform1 computation times. Where '**OpenCL**' denotes that all computations that can be accelerated are using the GPU; '**1 CPU core**' denotes that the PIV cross-correlation is performed in Java in the CPU with no OpenCL GPU acceleration and that PIV is running in single core mode; '**8 CPU core**' denotes that the PIV cross-correlation is performed in Java in the CPU with no OpenCL GPU acceleration, however PIV is running in multi-threaded mode using the all cores with a total of 8 threads; '**Mixed**' denotes that PIV cross-correlation is being accelerated by OpenCL, while the OpF methods are running in Java in the CPU. All execution times include the data exports and all computations required to process the given image database under the specified QuickLabPIV-ng processing configuration.

| Platform 1 - Ubuntu 20.04.6 LTS - Linux |                 |                |                 |                |
|---|-----------------|----------------|-----------------|----------------|
| Hybrid Configurations                   | Database1       |                | Database2       |                |
|   | Single pair (s) | 500 pairs (s)  | Single pair (s) | 500 pairs (s)  |
| Hybrid config. 1 - OpenCL               | 4.039           | 2019.468       | 4.925           | 2462.931       |
| Hybrid config. 2 - OpenCL               | 3.948           | 1974.008       | 4.810           | 2405.034       |
| Hybrid config. 3 - OpenCL               | 1.769           | 884.439        | 2.128           | 1063.992       |
| Hybrid config. 4 - OpenCL               | 1.703           | 851.484        | 1.978           | 989.039        |
| PIV Configurations                      | Database1       |                | Database2       |                |
|   | Single pair (s) | 500 images (s) | Single pair (s) | 500 images (s) |
| PIV config. 1 - OpenCL                  | 0.657           | 328.563        | 0.810           | 404.993        |
| PIV config. 2 - 1 CPU core              | 3.287           | 1643.273       | 3.936           | 1967.880       |
| PIV config. 3 - 8 CPU cores             | —               | —              | —               | —              |
| PIV config. 4 - OpenCL                  | 3.370           | 1684.857       | 3.078           | 1538.814       |
| PIV config. 5 - Mixed                   | 0.753           | 376.406        | 0.879           | 439.598        |

Table 4.2: Platform2 computation times in Linux. Where '**OpenCL**' denotes that all computations that can be accelerated are using the GPU; '**1 CPU core**' denotes that the PIV cross-correlation is performed in Java in the CPU with no OpenCL GPU acceleration and that PIV is running in single core mode; '**8 CPU core**' denotes that the PIV cross-correlation is performed in Java in the CPU with no OpenCL GPU acceleration, however PIV is running in multi-threaded mode using the all cores with a total of 8 threads; '**Mixed**' denotes that PIV cross-correlation is being accelerated by OpenCL, while the OpF methods are running in Java in the CPU. All execution times include the data exports and all computations required to process the given image database under the specified QuickLabPIV-ng processing configuration.

| Platform 2 - Ubuntu 20.04.6 LTS 64-bits - Linux |                 |                |                 |                |
|---|-----------------|----------------|-----------------|----------------|
| Hybrid Configurations                           | Database1       |                | Database2       |                |
|   | Single pair (s) | 500 pairs (s)  | Single pair (s) | 500 pairs (s)  |
| Hybrid config. 1 - OpenCL                       | 3.309           | 1654.731       | 4.044           | 2021.834       |
| Hybrid config. 2 - OpenCL                       | 3.266           | 1632.977       | 3.850           | 1925.027       |
| Hybrid config. 3 - OpenCL                       | 2.386           | 1193.012       | 2.878           | 1439.219       |
| Hybrid config. 4 - OpenCL                       | 2.229           | 1114.563       | 2.573           | 1286.352       |
| PIV Configurations                              | Database1       |                | Database2       |                |
|   | Single pair (s) | 500 images (s) | Single pair (s) | 500 images (s) |
| PIV config. 1 - OpenCL                          | 0.957           | 478.438        | 1,171           | 585.671        |
| PIV config. 2 - 1 CPU core                      | 8.142           | 4070.852       | 9.872           | 4935.980       |
| PIV config. 3 - 16 CPU cores                    | 2.814           | 1406.771       | 3.394           | 1697.221       |
| PIV config. 4 - OpenCL                          | 3.917           | 1958.726       | 3.647           | 1823.277       |
| PIV config. 5 - Mixed                           | 1.231           | 615.286        | 1.438           | 719.070        |

Table 4.3: Platform2 computation times in Windows. Where '**OpenCL**' denotes that all computations that can be accelerated are using the GPU; '**1 CPU core**' denotes that the PIV cross-correlation is performed in Java in the CPU with no OpenCL GPU acceleration and that PIV is running in single core mode; '**8 CPU core**' denotes that the PIV cross-correlation is performed in Java in the CPU with no OpenCL GPU acceleration, however PIV is running in multi-threaded mode using the all cores with a total of 8 threads; '**Mixed**' denotes that PIV cross-correlation is being accelerated by OpenCL, while the OpF methods are running in Java in the CPU. All execution times include the data exports and all computations required to process the given image database under the specified QuickLabPIV-ng processing configuration.

| Platform 2 - Microsoft Windows 10 64-bits - Windows |                 |                |                 |                |
|---|-----------------|----------------|-----------------|----------------|
| Hybrid Configurations                               | Database1       |                | Database2       |                |
|   | Single pair (s) | 500 pairs (s)  | Single pair (s) | 500 pairs (s)  |
| Hybrid config. 1 - OpenCL                           | 3.434           | 1717.154       | 4.104           | 2050.661       |
| Hybrid config. 2 - OpenCL                           | 3.299           | 1649.500       | 3.858           | 1928.830       |
| Hybrid config. 3 - OpenCL                           | 2.419           | 1209.446       | 2.920           | 1459.788       |
| Hybrid config. 4 - OpenCL                           | 2.293           | 1146.724       | 2.675           | 1337.554       |
| PIV Methods   | Database1       |                | Database2       |                |
|   | Single pair (s) | 500 images (s) | Single pair (s) | 500 images (s) |
| PIV config. 1 - OpenCL                              | 1.052           | 526.050        | 1.286           | 642.798        |
| PIV config. 2 - 1 CPU core                          | 8.582           | 4291.001       | 10.362          | 5181.219       |
| PIV config. 3 - 16 CPU cores                        | 2.826           | 1413.180       | 3.404           | 1702.081       |
| PIV config. 4 - OpenCL                              | 3.960           | 1980.171       | 3.741           | 1870.264       |
| PIV config. 5 - Mixed                               | 1.260           | 630.078        | 1.541           | 770.331        |

32 GB on Platform2, which can affect the processing times. This thread and memory pool limitation had to be applied due to the total system memory available not being enough in that platform. Still Platform2 despite having a higher core count has a much older CPU with a lower base frequency. In any case we see that the results don't change too much between both platforms and that QuickLabPIV-ng can achieve sub-second times for some of the configurations even with the higher resolution images, which can be considered near real-time performance. In Platform2 the Windows and Linux performance is almost the same, with a slight advantage for Linux, which demonstrates the true multi-platform behavior of QuickLabPIV-ng. When looking at the computation times between PIV configuration 4 and PIV configuration 5, it can be seen that not always is good to have all computations to be OpenCL accelerated. In fact when using OpF for the PIV sub-pixel with PIV performing vector validation and substitution, it is preferable to employ Java CPU only OpF sub-pixel, because the OpenCL is only good for massively parallelizing massive amounts of similar operations. However, in this case the operations are done with the individual IA sub-pixel values, which can change as individual IA vector validation and substitution is applied, thus resulting in higher processing times. It makes no sense to process every single pixel of the image to just update a single vector. It is also visible by comparing the computation times between Database1 and Database2, that the increase of the image resolution within reasonable limits does not cause the computation times to grow exponentially for any of the PIV/Hybrid PIV computation methods. For Platform2 it becomes apparent that the PIV cross correlation can be more efficiently computed in the GPU by a factor of 2, even when utilizing all CPU cores for the PIV computation.

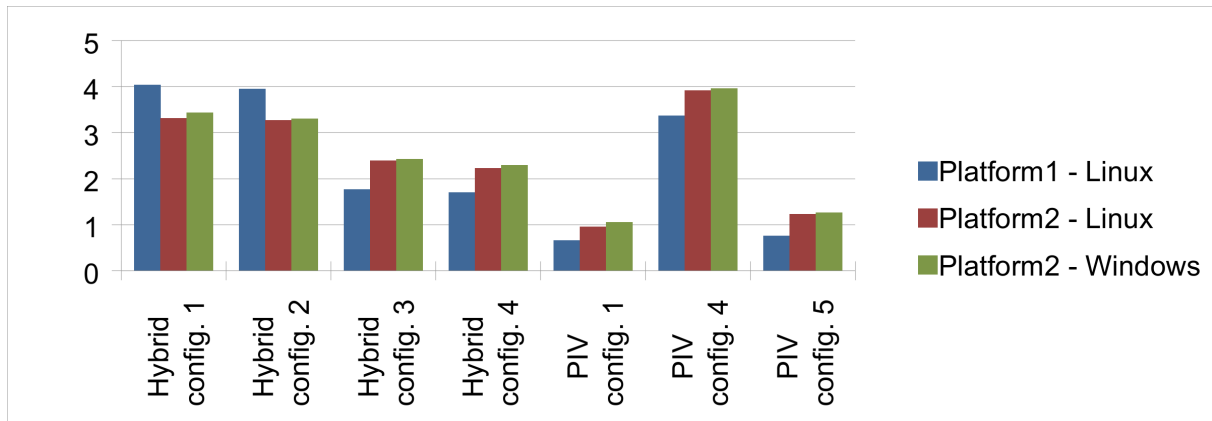


Figure 4.5: QuickLabPIV-ng computational times comparison for Database1. Database2 relative performance between methods and platforms would be similar to Database1 and is not shown.

The Figure 4.5 illustrates that the export times between sparse and dense data exports are almost identical compared to the total processing times as seen by comparing Hybrid config. 1 with Hybrid config. 2 and when comparing Hybrid config. 3 with Hybrid config. 4. It also visible that applying PIV with Lucas-Kanade sub-pixel as in configuration PIV config. 5 adds little overhead with respect to PIV config. 1 with Howngwei Guo sub-pixel.

## 4.5 References

- [1] Raffel, M.; Willert, C.; Wereley, S.; Kompenhans, J. Particle Image Velocimetry—A Practical Guide; Springer: Berlin/Heidelberg, Germany, 2007. <https://doi.org/10.1007/978-3-540-72308-0>.
- [2] Guo, H., A Simple Algorithm for Fitting a Gaussian Function. In Streamlining Digital Signal Processing; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2012; Chapter 31, pp. 297–305. <https://doi.org/https://doi.org/10.1002/9781118316948.ch31>.
- [3] Mendes, L.P.N.; Ricardo, A.M.C.; Bernardino, A.J.M.; Ferreira, R.M.L. A comparative study of optical flow methods for fluid mechanics. Exp. Fluids **2021**, *63*, 7. <https://doi.org/10.1007/s00348-021-03357-7>.
- [4] Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI'81, Vancouver, BC, Canada, 24–28 August 1981; Volume 2, pp. 674–679.
- [5] Liu, T.; Salazar, D. OpenOpticalFlow\_PIV: An Open Source Program Integrating Optical Flow Method with Cross-Correlation Method for Particle Image Velocimetry. J. Open Res. Softw. **2021**, *9*, 3. <https://doi.org/10.5334/jors.326>.
- [6] Gamma, E., Helm, R., Johnson, R.E., Vlissides, J. Design patterns: elements of reusable object-oriented software. Addison-Wesley Professional.

## Chapter 5

# A Hybrid PIV/Optical Flow Method for Incompressible Turbulent Flows

Article published in the MDPI Water journal with DOI: <https://doi.org/10.3390/w16071021>

Received: 19 February 2024, Revised: 20 March 2024, Accepted: 26 March 2024, Published: 1 April 2024.

The article was split into two chapters, this one being the second part. All paper pre-existing sections are left as in the original published paper.

### 5.1 Introduction

Particle Image Velocimetry (PIV) has become the standard for the non-intrusive assessment of kinematic quantities in industry as well as in applied and fundamental research. Planar PIV provides two or three dimensional velocity fields, the latter with a stereoscopic camera arrangement. Three-dimensional PIV, currently dominated by tomographic PIV, provides all three components of the velocity in volumes normally illuminated by expanded laser beams [1]. It is a relevant tool for environmental or ecological studies involving fluid mechanics, as well as aerospace, the automotive industry, civil engineering, hydraulics, fluid mechanics, and even medical and veterinary sciences. Example applications include pollutant dispersion, pollutant emission reduction, aerodynamics (planes, vehicles, or buildings), and turbulent flow analysis, as well as floods, or even blood flow studies of arterial blood vessels.

PIV methods typically assume that the region of interest is divided into small areas or volumes (interrogation areas/units—IAs), where the flow is approximately uniform. The motion of a set of particles in the interrogation units is then converted into a velocity vector through cross-correlation and sub-pixel analysis (see Figure 5.1). The size of the interrogation units determines the spatial resolution of PIV. Small units improve the resolution but reduce the signal quality, as the method requires several particles in the interrogation unit to obtain a strong correlation peak. Finding the correlation peak with sub-pixel accuracy is fundamental for the overall accuracy of the method. In “adaptive” or “multi-pass” PIV, this workflow is cycled several times, where, at each cycle/step, the interrogation units are refined to

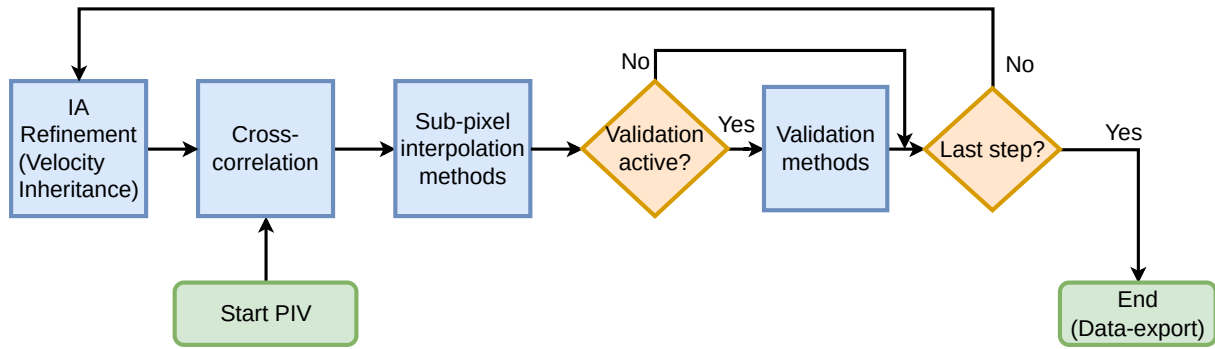


Figure 5.1: Classic PIV processing steps/workflow.

smaller sizes, with the number of steps/cycles being predefined by the user. The velocity vectors of each interrogation unit, from the previous step, are inherited by the newly refined units, according to a given inheritance strategy.

The resolution of classic PIV does not increase with the increase in tracer particles in the interrogation units.

Particle Tracking Velocimetry (PTV) can provide a velocity vector per tracer particle [2, 3], greatly increasing the spatial resolution of classic PIV. The accuracy of Particle Tracking Velocimetry depends on the correct identification of the same particle between consecutive images and on the identification of its centroid with sub-pixel resolution. It is normally not practical for planar (2D) PIV since the out-of-plane loss of particles may reduce its advantages vis-a-vis the added computational cost. It also requires interpolation into regular grids if subsequent analysis requires evenly spaced data.

A different velocimetry technique that may provide denser 2D vector maps is optical flow (OpF). Optical flow estimates the apparent 2D flow resulting from time-varying image brightness intensity through optimization techniques. OpF methods can be classified as gradient-based, e.g., [4–7], region-based or feature-based matching [8–10], global-matching [11, 12], spatio-temporal filtering [13, 14], and, more recently, deep learning methods [15, 16]. Of the many OpF methods that have been developed since their inception, most of them have specifically targeted machine vision applications; however, fluid mechanics applications started to be explored in the 1990s. The first OpF method for fluid mechanics was probably [17], featuring a global OpF method and a dynamic programming class algorithm. Other attempts at bringing OpF to the hydraulics and fluid mechanics fields are based on the Horn and Schunck [4] method. This is the case for [18] and several variants with different computational constraints [19–22]. A few other modern methods developed for fluid mechanics applications include FOLKI [23, 24], Liu–Shen physics-based OpF [25], and wOFV [26, 27]. Theoretically, dense OpF methods can achieve one vector per image pixel. However, the relatively sparse (identifiable) bright dots of PIV images may not allow for such high resolution.

Hybridization is a way to take advantage of favorable image characteristics, such as higher tracer concentrations, or hardware availability, such as several visualization angles by several synchronized cameras, to increase spatial resolution. Few attempts have been made at hybridizing PIV with OpF techniques. The most relevant ones include [28], with the proposal of a combination of PIV with the



OpF method of Liu and Shen [25], the latter without employing any of the traditional PIV sub-pixel methods. Glomb and Swirniak (2019) [29] presented methods for hybrid PIV-OpF, as well as hybrid Particle Tracking Velocimetry (PTV), based on PIV and multi-resolution/pyramidal OpF. Seong (2019) [30] conceived a hybrid PIV-OpF approach with a Horn–Schunck-inspired method [31] that was adapted to deal with variations in laser intensity for pulsed laser configurations. The works [32, 33] also developed a hybrid PIV-OpF solution that addresses image intensity variations between successive frames. It uses PIV in connection with Liu–Shen OpF [25] and with Horn–Schunck [4] OpF, where the latter provides the initial estimate for the former method. Hybrid methods have the ability to deal with large tracer particle displacements while retrieving dense velocity maps like dense OpF methods. Some hybrid PIV solutions do not require multi-resolution/pyramidal OpF setups to deal with the large displacements, for example, [28]. The novel hybrid PIV method in [34] explores a different approach to computing the cross-correlation by considering a circulant matrix and also includes a novel OpF method inspired by the Horn–Schunck formulation that better handles large displacements and that dynamically adjusts its parameters.

In summary, previous PIV-OpF hybridization efforts have attempted to streamline procedures, provide efficient implementations, test novel combinations of methods, or even present novel methods. However, there are many possible hybrid combinations of correlation-based PIV and OpF methods, mostly because there are several available OpF methods. Recently, the study [35] extensively analyzed various combinations of OpF methods and showed that the combination of the Lucas–Kanade and Liu–Shen OpF methods is the most accurate. To our best knowledge, this combination has not been hybridized with PIV. The first objective of this paper is thus to present a hybrid of correlation-based PIV and a combination of Lucas–Kanade and Liu–Shen OpF methods. We present a novel open-source tool, named QuickLabPIV-ng (<https://github.com/CoreRasurae/QuickLabPIV-ng>, accessed on 29 March 2024), that supports hybrid PIV with a combination of the Liu–Shen and Lucas–Kanade OpF methods [35], where the latter provides the initial estimate for Liu–Shen.

The second objective of this paper is to explore the accuracy of a hybrid PIV method that replaces sub-pixel interpolation with an optical flow (OpF) step to refine the estimation of displacement within each interrogation area. This method aims to address issues related to the size and number of particle images affecting the correlation peak’s shape, which can lead to errors in displacement estimation. By integrating OpF, the process seeks to directly correct the peak location’s integer estimate, potentially circumventing the challenges associated with traditional sub-pixel reconstruction steps in PIV analysis.

Hybrid PIV-OpF methods have been validated against well-known phenomena but not rigorously tested on real turbulent flows critical for environmental studies. Recognizing the importance of denser velocity maps for analyzing turbulence [36], this paper proposes testing these methods on realistic flows resembling homogeneous isotropic turbulence. Additionally, we highlight the limitations of applying OpF to datasets designed for classic PIV and set out to validate hybrid solutions on real-life turbulent flow data with inherent optical noise, marking a significant step toward practical application.

Finally, we validate the new hybrid PIV method and the sub-pixel OpF-based alternative on synthetic and real laboratory PIV image databases. On the one hand, validation on the synthetic databases is

aimed at evaluating the raw accuracy performance by validating the methods against precise ground-truth data, as generated by the PIV image generator tool [37] (please see section 5.3 for further details about the image generation). On the other hand, validation on real laboratory databases is aimed at evaluating the performance of the methods with respect to turbulent macro- and micro-scale recovery.

This paper is organized as follows. It starts with a review of the proposed PIV software and techniques. It then validates these methods using synthetic images of simple flows as a baseline, assessing errors across various noise levels. The second validation phase applies the software to real experimental data of differing qualities (in terms of image noise, non static background image elements, and varying imaged particle's sizes and concentrations), including flows around a cylinder, in a rough boundary layer, and plunging flows. The results are analyzed using auto-correlation, structure functions, power spectral density, and dissipation rates. The paper concludes with findings and recommendations for future research.

## 5.2 Verification and Validation of the Hybrid PIV-OpF Software

### 5.2.1 Strategy

In this section, we evaluate the performance of QuickLabPIV-ng PIV on synthetic and real datasets according to the three variants described in the previous section. The evaluation of synthetic data performance focuses on accuracy and spatial error distribution compared to the ground truth across synthetically generated datasets. The evaluation uses a PIV image generator tool [37], enhanced in version 1.2.0 to differentiate between wall and flow regions [40], exemplified in [35] in a Poiseuille flow scenario. The methodology includes analyzing a Statistical Image Group (SIG) comprising images with varying parameters like flow type, maximum velocity, bit depth, white noise variance, and particle size and concentration. The evaluation calculates both average relative and absolute errors across the entire SIG, excluding vectors with no ground-truth velocity. Additionally, it assesses in-plane ( $F_i$ ) and out-of-plane ( $F_o$ ) particle losses for PIV methods. For hybrid PIV methods, in-plane losses cannot be measured due to the absence of interrogation areas in the final step, and out-of-plane losses are estimated using the standard deviation of movement. Although the study tested various bit depths, the discussion focuses on 8-bit results due to their similarity.

For real experimental data, the analysis focuses on power spectral density, structure functions, and auto-correlation functions using laboratory PIV data. This qualitative assessment aims to determine whether hybrid PIV, utilizing either the Lucas–Kanade method alone or in combination with the Liu–Shen method, enhances the effective measurement range over standard PIV methods. This includes evaluating improvements in capturing the auto-correlation, power spectral density, and 2nd-order structure function of velocity fluctuations. Hybrid PIV's ability to analyze down to individual pixels could potentially expand the data range from regular PIV datasets, especially in turbulent flows and across varying image qualities.

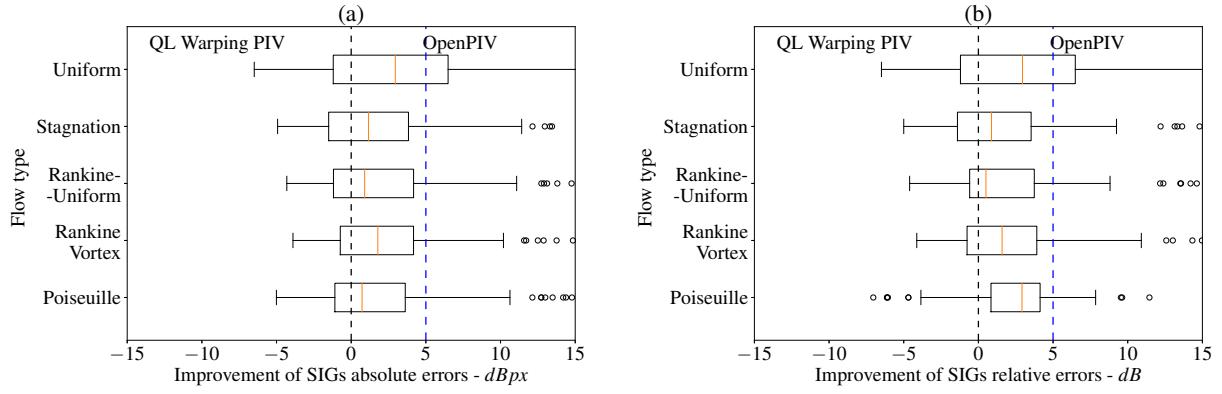


Figure 5.2: An illustration of the relative accuracy performance between QuickLabPIV-ng (QL)’s Modern PIV (micro-warping) and OpenPIV’s Modern PIV with respect to the absolute and relative errors. Absolute errors are shown in sub-figure (a), and relative errors are in sub-figure (b). The vertical black dashed line at the center is the null-advantage line, data to the left indicate a QuickLabPIV-ng advantage, and data to the right favor the OpenPIV method. The rectangles on the left side and right side denote the Q1 quartile and Q3 quartile, respectively. The vertical red line denotes the median, or the Q2 quartile, while the right whisker denotes the Q4 quartile. The circles denote outliers that are outside the margin defined by three times the inter-quartile range (IQR), centered around the mean of the distribution, where the IQR is the difference between Q3 and Q1. The advantage is determined by the difference in relative accuracy between the two methods for all the considered parameters. The vertical blue dashed line marks the 5 dB advantage for OpenPIV.

## 5.2.2 Synthetic Data

### Comparison of Correlation-Based PIV Implemented in QuickLabPIV-ng with State-of-the-Art PIV

We tested five flow types, namely, uniform flow, inviscid flow with a stagnation point, Rankine vortex, Rankine vortex with superimposed uniform flow, and planar Poiseuille flow. All flows were generated for each combination of particle spot diameter (1.0, 2.0, 3.0, and 6.0 pixel spot sizes); White Gaussian Image Noise (WGIN) (0.0, 5.0, and 15.0 dB); maximum velocity displacement vector (0.8, 1.6, and 4.0 px/frame); particle concentration per IA volume (1, 6, 12, and 16 particles/IA volume); and standard out-of-plane movement (0.025, 0.050, and 0.100 mm).

The traditional PIV of QuickLabPIV-ng’s modern PIV approach was evaluated against the state-of-the-art OpenPIV, at version 0.25.0. OpenPIV applies warping to each IA pair, followed by a cross-correlation of each warped IA pair and ending with a sub-pixel method. OpenPIV also includes a bi-cubic spline based velocity filter/smoothness method and integrates vector validation techniques.

Figure 5.2 provides an overview of the relative accuracy for the traditional PIV mode between QuickLabPIVng and OpenPIV. As we can see, considering the median, QuickLabPIV-ng is within 3 dB of OpenPIV, with the latter having a slight general advantage.

### Accuracy of QuickLabPIV-ng by Flow Type

In this sub-section, we analyze several performance aspects, accuracy-wise, with respect to the synthetic image databases. We start with an assessment of the relative performance between warping PIV in QuickLabPIV-ng with respect to the well-known and state-of-the-art OpenPIV with regard to the non-hybrid PIV modes; see Figure 5.2. It is not our intent to benchmark this tool against OpenPIV, or

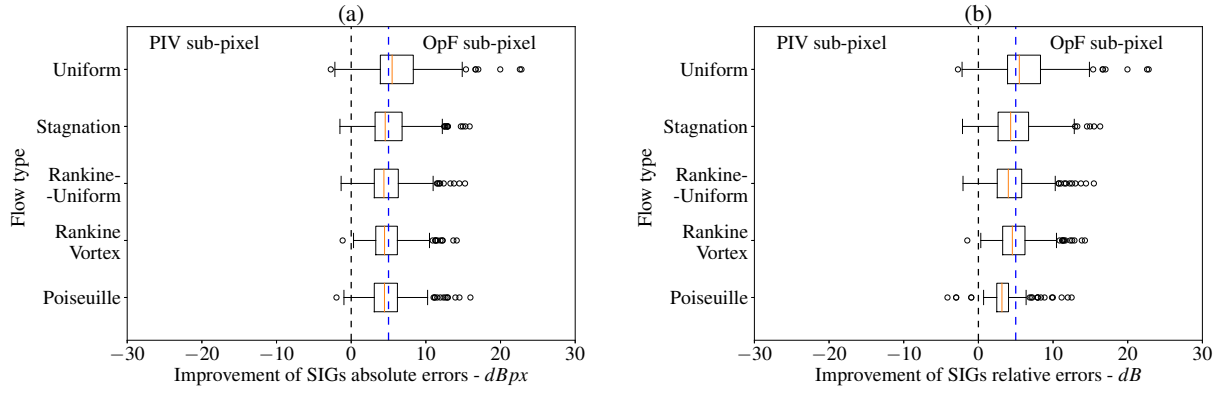


Figure 5.3: The advantage of a last step of sparse the OpF sub-pixel with Liu–Shen combined with Lucas–Kanade over standard sub-pixel interpolation. The vertical blue dashed line marks the 5 dB advantage for the OpF sub-pixel (variant 3). Sub-figure (a) refers to absolute errors, while sub-figure (b) refers to relative errors. For further details, please refer to Figure 5.2 legend.

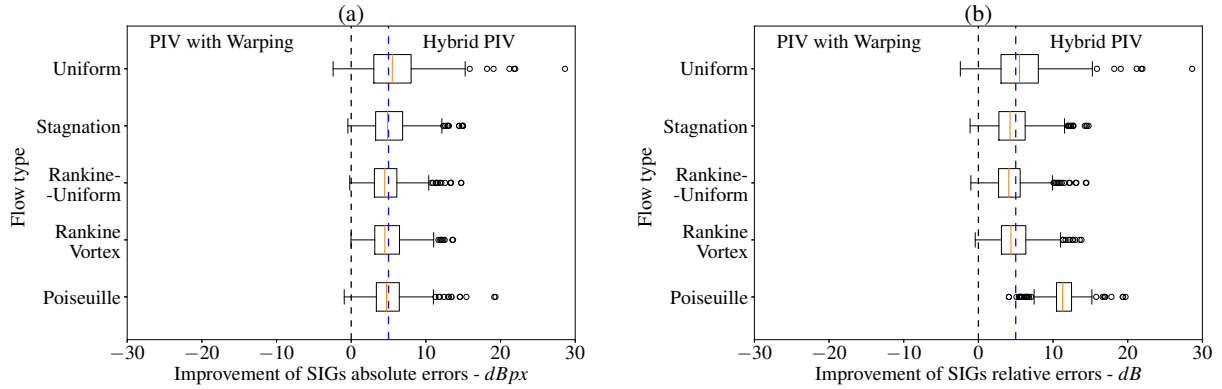


Figure 5.4: The advantage of hybrid PIV to standard PIV with warping. The vertical blue dashed line marks the 5 dB advantage for the hybrid PIV mode (variant 1). Sub-figure (a) refers to absolute errors, while sub-figure (b) refers to relative errors. For further details, please refer to Figure 5.2 legend.

other any other PIV tool; we just want to show that QuickLabPIV-ng’s base PIV with warping is mostly in line with OpenPIV’s performance.

Figure 5.3 evidences the accuracy advantage of employing a final step of sparse OpF with Liu–Shen combined with Lucas–Kanade, for all the considered flows, with respect to the standard sub-pixel employed in standard PIV with warping.

In Figure 5.4, it can be seen that hybrid PIV slightly improves the advantage observed for the OpF sub-pixel in standard PIV (compared with Figure 5.3) for all the flows. The Poiseuille flow has a more pronounced advantage for hybrid PIV, which is due to the smaller proportion of highly erroneous vectors to the total number of velocity vectors.

### Overall (All Flows) Accuracy Span

Figure 5.5 depicts the results averaged across all the flow types for the three methods—PIV with warping, PIV with warping combined with a final OpF sub-pixel, and hybrid PIV. The OpF sub-pixel improves the accuracy but also increases the variance in the relative error, particularly for the smallest particle spot sizes. The introduction of hybrid PIV techniques yields dense data with comparable accu-

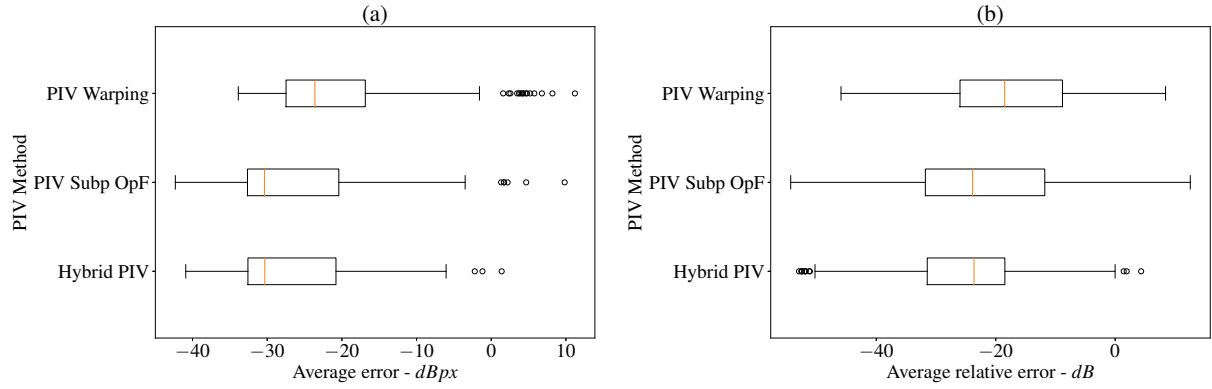


Figure 5.5: An overview of the overall accuracy distribution for the QuickLabPIV-ng method type for all the considered flow types. In sub-figures (a,b), we see the average of the absolute errors in a SIG and the average of the relative errors in a SIG, respectively. The PIV with warping method is PIV with micro-warping and Hongwei Guo’s 1D-1D robust linear regression sub-pixel with a 5 px width and 20 iterations and no velocity vector validation. The PIV Subp OpF method is identical to the PIV warping method, with the exception that Hongwei Guo’s sub-pixel is parameterized with a 3 px width while keeping the same number of iterations, all followed by a final step with sparse optical flow with the Liu–Shen method combined with the Lucas–Kanade method. The hybrid PIV method is identical to the PIV Subp OpF method, with the exception that the final optical flow is computed in dense mode, that is, one velocity vector per image pixel. For further details, please refer to Figure 5.2 legend.

racy to PIV enhanced by the OpF sub-pixel while also reducing the variance observed in relative errors. This indicates a balance between improved detail and the stability of measurement outcomes.

Figure 5.6 summarizes the absolute error versus effective particle concentration per IA volume with respect to different particle sizes and WGIN noise levels. A PIV method with micro-warping (left) and two hybrid PIV methods, one with Lucas–Kanade OpF (middle) and another with Liu–Shen combined with Lucas–Kanade OpF (right), are evaluated. The relative errors are not shown since the performance is similar for this flow. Figure 5.7 rearranges the data to compare relative errors against particle spot sizes and flow types with a constant displacement of 4.0 px/frame. It assesses the average relative error across various WGIN noise levels for the three PIV methodologies: a PIV method with micro-warping (left) and two hybrid PIV methods, one with Lucas–Kanade OpF (middle) and another with Liu–Shen combined with Lucas–Kanade OpF (right). The absolute errors are not shown since the performance is similar.

The findings in Figures 5.6 and 5.7 suggest avoiding 1.0 px particle sizes across all methods and generally steering clear of low concentrations, like 1 particle per IA, except possibly in hybrid PIV with larger particles and minimal noise. The Poiseuille flow posed the greatest challenge, unlike the more straightforward uniform flow. Accuracy did not significantly vary with particle sizes of 3.0 px and above or concentrations over 6 particles per IA in OpF methods. The optimal results were seen with 16 particles per IA and 3.0 px sizes, though larger 6.0 px particles showed reduced variance, not necessarily improving accuracy.

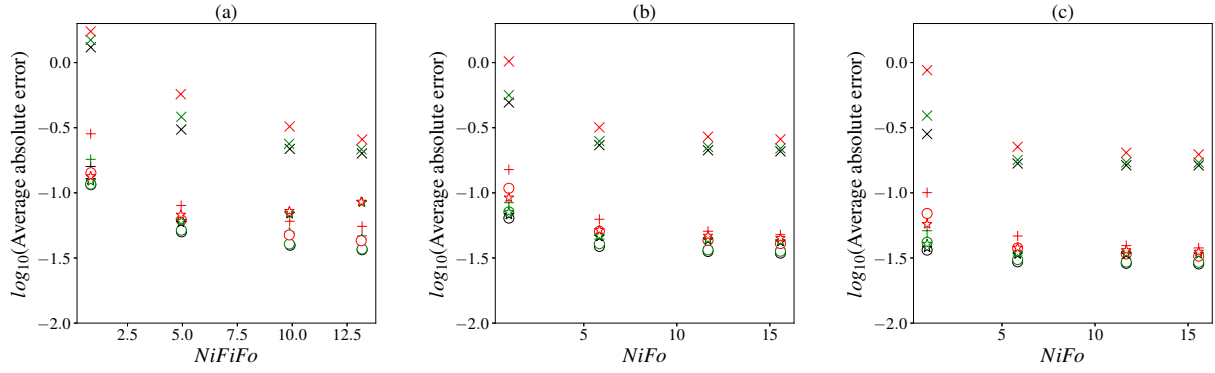


Figure 5.6: The figures show the absolute error versus actual particle concentration per IA volume ( $N_i F_i F_o$ ) or by image for hybrid PIV ( $N_i F_o$ ), since the latter has no concept of IA. The data pertain to the Rankine vortex with superimposed uniform flow, 4.0 px maximum velocity vector magnitude, and 8-bit depth, with four values of number particle concentration  $N_i$ , namely, 1, 6, 12, and 16 particles per IA volume, and three levels of WGIN noise, 0, 5, and 15 dBW. Sub-figure (a) is the PIV method with micro-warping and Hongwei Guo's robust linear regression of 1D-1D Gaussian as the sub-pixel method with 5.0 px and 20 iterations; in sub-figure (b), we have the hybrid PIV method with micro-warping employing the same Hongwei Guo's sub-pixel with a 3.0 px width and 20 iterations, followed by a final step with Lucas-Kanade optical flow; in sub-figure (c), we have the hybrid PIV method with micro-warping employing the same Hongwei Guo's sub-pixel with a 3.0 px width and a final step with Liu-Shen combined with Lucas-Kanade optical flow. (Legend:  $\times$ —denotes 1 px diameter imaged particle spot size;  $+$ —denotes 2 px diameter of imaged particle spot size;  $\bigcirc$ —denotes 3 px diameter imaged particle spot size;  $\star$ —denotes 6 px diameter imaged particle spot size; Black—denotes 0 dBW WGIN; Green—denotes 5 dBW WGIN; Red—denotes 15 dBW WGIN.)

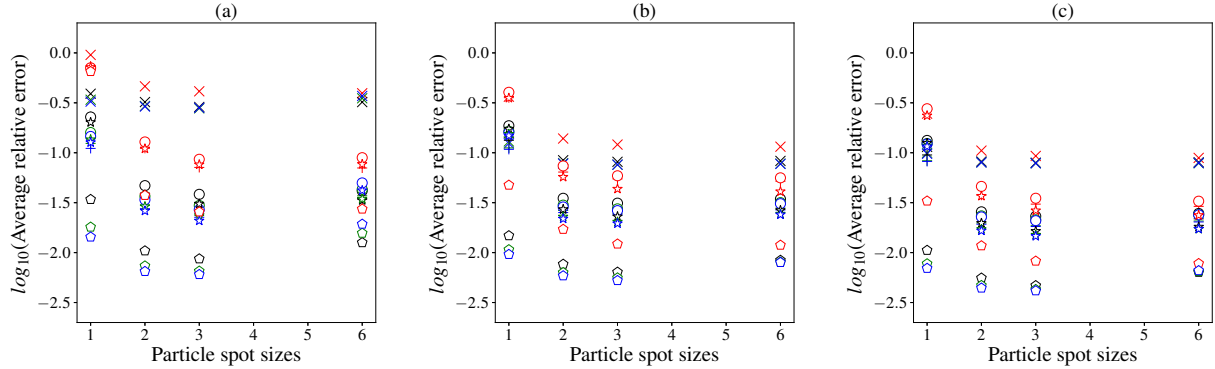


Figure 5.7: All flows compared with respect to relative errors, with a 4.0 px maximum velocity vector magnitude and an 8-bit depth with the average noise level. PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 5.0 px and 20 iterations is shown in sub-figure (a). In sub-figure (b), we have hybrid PIV with micro-warping employing the same Hongwei Guo's sub-pixel and a final step with Lucas-Kanade optical flow. Finally, in sub-figure (c), we have hybrid PIV with micro-warping employing the same Hongwei Guo's sub-pixel and a final step with Liu-Shen combined with Lucas-Kanade optical flow. (Legend:  $\times$ —denotes a Poiseuille flow;  $+$ —denotes the Rankine vortex;  $\bigcirc$ —denotes the Rankine vortex with a superimposed uniform flow;  $\star$ —denotes an inviscid flow with a stagnation point;  $\diamond$ —denotes a uniform flow; Red—denotes 1 particle per IA volume; Black—denotes 6 particles per IA volume; Green—denotes 12 particles per IA volume; Blue—denotes 16 particles per IA volume.)

## Spatial Error Distribution

This study examines the spatial distribution of absolute and relative errors across various flow types: Poiseuille, Rankine vortex, Rankine vortex combined with uniform flow, inviscid flow with a stagnation point and uniform flow. All flows have 3.0 px diameter particles with a concentration of 12 particles per IA, a maximum velocity of 4.0 px/frame, and a WGIN noise level of 0 dBW. The evaluation encompasses three methods: micro-warping PIV; micro-warping PIV enhanced with a sparse sub-pixel approach based on Liu–Shen combined with Lucas–Kanade OpF in the final step; and dense hybrid PIV utilizing Liu–Shen with Lucas–Kanade OpF. These methods, which do not include vector validation, were assessed using adaptive PIV strategies, specific pixel width settings for the sub-pixel analysis, and distinct iterations and window sizes for the Lucas–Kanade and Liu–Shen algorithms, highlighting the performance of each configuration under the given conditions.

In the Poiseuille flow analysis (Figure 5.8), the highest inaccuracies were noted near the horizontal walls at the top and bottom. The micro-warping PIV method exhibited the largest average error in both relative and absolute terms, surpassed by the sparse OpF sub-pixel PIV in error magnitude. The hybrid PIV method demonstrated superior performance, showing a significantly smaller proportion of highly erroneous vectors (Table 5.1). Notably, the most significant errors in the hybrid PIV method were confined to narrow regions near the top and bottom, each about 1 px in width.

Table 5.1: Averaged absolute and relative errors for the Poiseuille flow for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume.

| Method Name  | Absolute Error (Mean) | Relative Error (Mean) |
|--|-----------------------|-----------------------|
| PIV warping  | −26.886 dBpx          | −11.132 dB            |
| PIV warping, final sub-pixel                           | −30.160 dBpx          | −12.426 dB            |
| Liu–Shen combined with<br>Lucas–Kanade                 |                       |                       |
| Hybrid PIV with Lucas–Kanade                           | −28.346 dBpx          | −22.598 dB            |
| Hybrid PIV with Liu–Shen<br>combined with Lucas–Kanade | −30.400 dBpx          | −22.278 dB            |

In Figure 5.9, the Rankine vortex presents two challenging areas: one involving the absolute error at the boundary between the forced and free vortex regions, and another concerning the relative error at the vortex center where the velocity is zero. Table 5.2 illustrates that the OpF sub-pixel enhances accuracy compared to standard warping PIV. The hybrid method, equivalent to the OpF sub-pixel, maintains the mean accuracy while offering detailed velocity maps. Hybrid Lucas–Kanade achieves similar accuracy to warping PIV but with the added benefit of denser velocity data.

The Rankine vortex combined with a uniform flow shows similar patterns of absolute errors to the Rankine vortex alone but with reduced severity due to the velocity division between the vortex and uniform flow. In terms of relative errors, two zero-velocity regions pose challenges (Figure 5.10). Overall, across the entire image area, the trend observed with the Rankine vortex persists, indicating that hybrid PIV maintains a comparable mean accuracy to standard PIV methods, as highlighted in Table 5.3.

For the inviscid flow with a stagnation point (Figure 5.11 and Table 5.4), the relative errors are low across all methods, making them less noteworthy. The distinction comes with the absolute errors, which

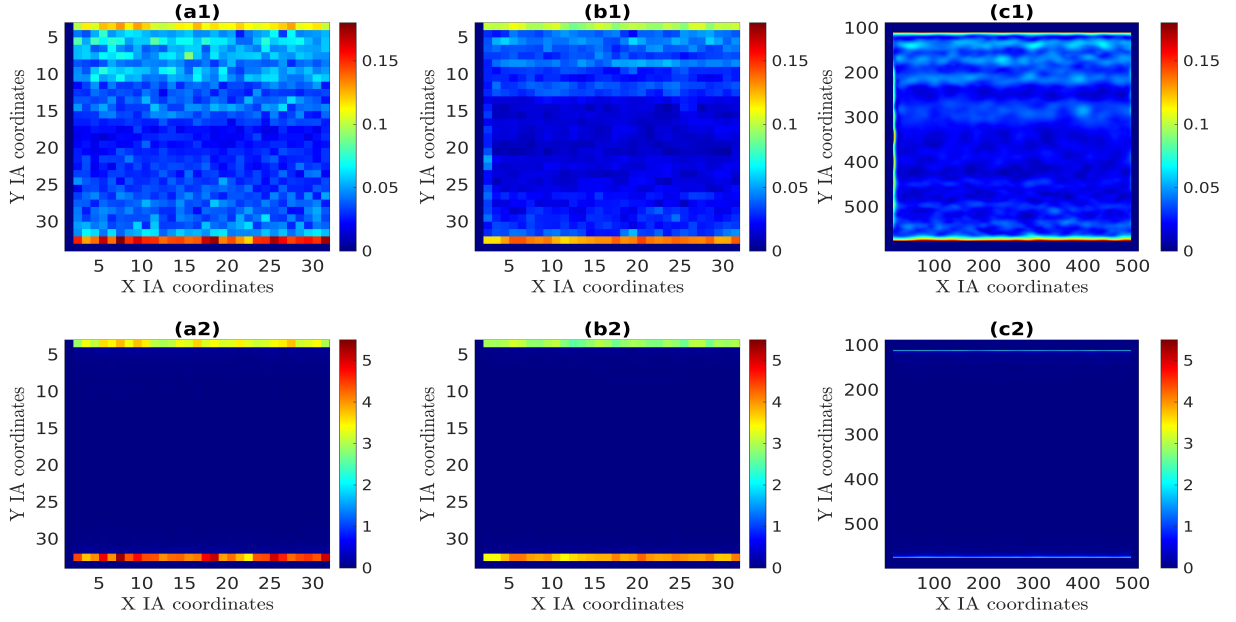


Figure 5.8: Spatial distributions of absolute and relative errors represented through Figures (x1) and Figures (x2), respectively, where x is either **a**, **b**, or **c**. These figures are for the Poiseuille flow, with a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures (ax), where x is either **1** or **2**, present velocimetry data obtained by PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 5.0 px and 20 iterations. Figures (bx), where x is either **1** or **2**, present velocimetry data obtained by PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 3.0 px and 20 iterations followed by a last step with sparse optical flow with Liu–Shen combined with Lucas–Kanade. Figures (cx), where x is either **1** or **2**, present velocimetry data obtained by hybrid PIV with micro-warping and Hongwei Guo's robust linear regression of 1D Gaussian as the sub-pixel method with 3.0 px and 20 iterations followed by a last step with dense optical flow with Liu–Shen combined with Lucas–Kanade. The data shown are the mean relative or mean absolute errors of a SIG for each IA (for PIV methods) or each pixel (for hybrid PIV).

exhibit similar patterns for both PIV and hybrid PIV to those seen in other flow types. Specifically, PIV enhanced with Liu–Shen and Lucas–Kanade OpF outperforms standard PIV, and when this combination is applied in hybrid PIV, it achieves comparable accuracy with the benefit of denser velocity maps. Similarly, hybrid PIV using dense Lucas–Kanade matches the accuracy of standard PIV, also providing denser velocity data.

With respect to the uniform flow, we see that the best results are obtained when the Liu–Shen combined with Lucas–Kanade is employed, as documented by Figure 5.12 and Table 5.5, both for PIV with OpF sub-pixel and dense hybrid PIV modes. The relative results are not shown since the error distribution is similar to the absolute errors. The uniform flow achieves the best accuracy of all the considered flows, posing the least challenge for all of the algorithms considered.



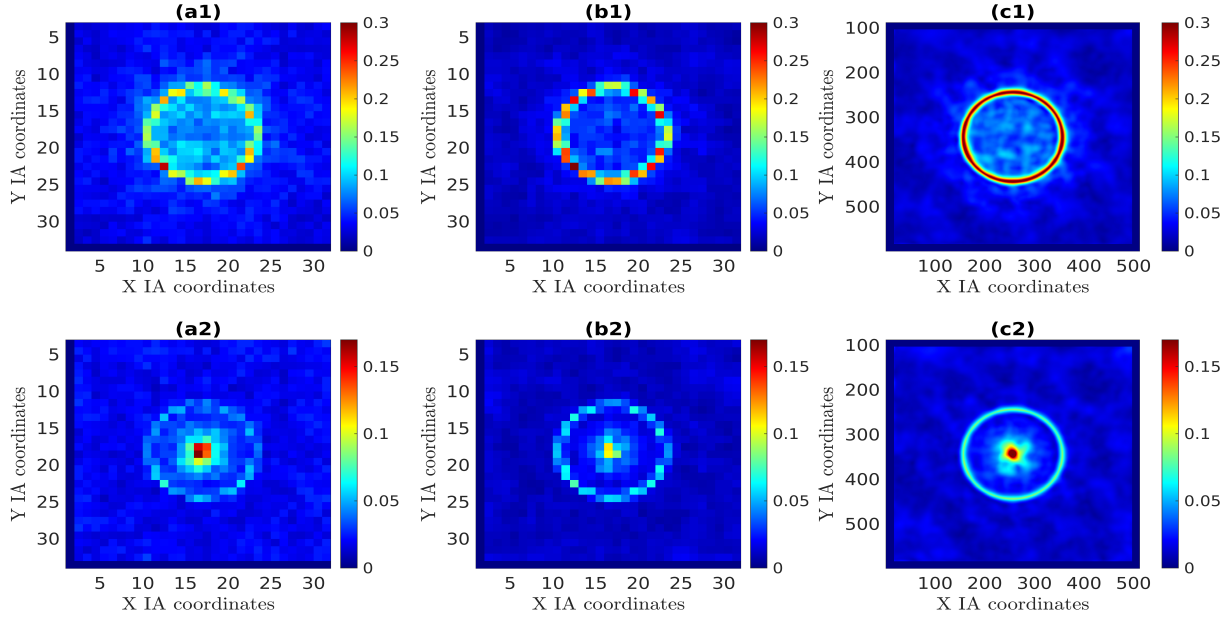


Figure 5.9: Spatial distribution of absolute and relative errors represented through Figures (x1) and Figures (x2), respectively, where x is either a, b, or c. These figures are for the Rankine vortex, with a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures (ax), (bx) and (cx), where x is either 1 or 2, represent the same parameterizations as in Figure 5.8.

Table 5.2: Averaged absolute and relative errors for the Rankine vortex for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume.

| Method Name   | Absolute Error (Mean) | Relative Error (Mean) |
|---|-----------------------|-----------------------|
| PIV warping   | −25.782 dBpx          | −32.692 dB            |
| PIV warping, final sub-pixel                        | −29.169 dBpx          | −36.624 dB            |
| Liu–Shen combined with Lucas–Kanade                 |                       |                       |
| Hybrid PIV with Lucas–Kanade                        | −25.763 dBpx          | −32.636 dB            |
| Hybrid PIV with Liu–Shen combined with Lucas–Kanade | −27.765 dBpx          | −35.063 dB            |

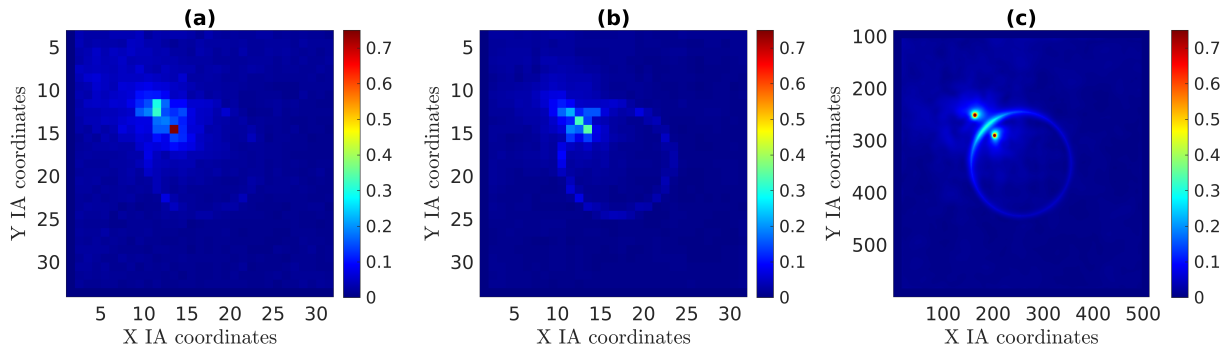


Figure 5.10: The spatial distribution of relative errors represented through Figures a, b, or c. The figures for the absolute errors are not shown, as they are mostly identical to their equivalents in Figure 5.9. These figures are for the Rankine vortex with a superimposed uniform flow, having a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures (a–c) have the same method configurations as (a2–c2) in Figure 5.8.

Table 5.3: Averaged absolute and relative errors for the Rankine vortex with a superimposed uniform flow for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume.

| Method Name   | Absolute Error (Mean) | Relative Error (Mean) |
|---|-----------------------|-----------------------|
| PIV warping   | −28.046 dBpx          | −31.206 dB            |
| PIV warping, final sub-pixel                        | −31.808 dBpx          | −35.069 dB            |
| Liu–Shen combined with Lucas–Kanade                 |                       |                       |
| Hybrid PIV with Lucas–Kanade                        | −29.005 dBpx          | −31.880 dB            |
| Hybrid PIV with Liu–Shen combined with Lucas–Kanade | −30.826 dBpx          | −33.986 dB            |

Table 5.4: Averaged absolute and relative errors for the inviscid flow with a stagnation point for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume.

| Method Name   | Absolute Error (Mean) | Relative Error (Mean) |
|---|-----------------------|-----------------------|
| PIV warping   | −29.612 dBpx          | −33.480 dB            |
| PIV warping, final sub-pixel                        | −34.359 dBpx          | −38.359 dB            |
| Liu–Shen combined with Lucas–Kanade                 |                       |                       |
| Hybrid PIV with Lucas–Kanade                        | −30.864 dBpx          | −34.673 dB            |
| Hybrid PIV with Liu–Shen combined with Lucas–Kanade | −33.545 dBpx          | −37.249 dB            |

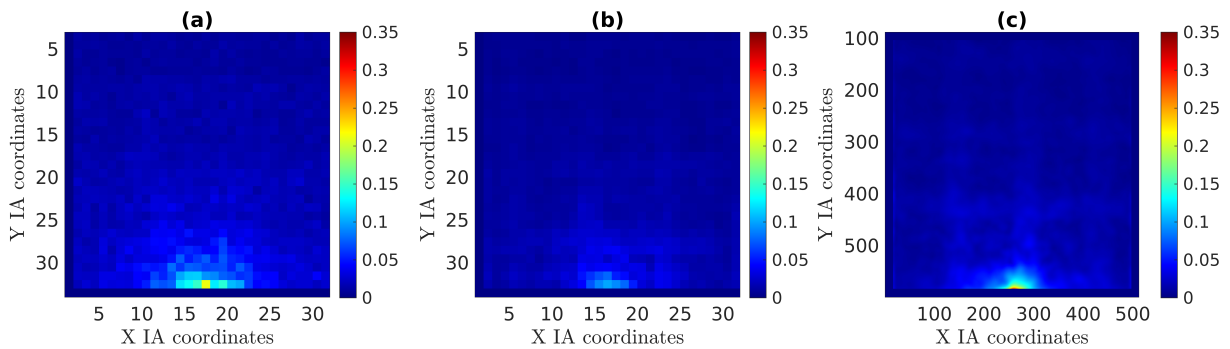


Figure 5.11: Spatial distribution of relative errors represented through Figures **a**, **b**, or **c**. Absolute errors are not shown since they are essentially low everywhere. These figures are for the inviscid flow with a stagnation point, having a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures **(a–c)** have the same method configurations as **(a2–c2)** in Figure 5.8.

Table 5.5: Averaged absolute and relative errors for a uniform flow for 4.0 px max. velocity and 3.0 px particle spot sizes, 0 dB WGIN, and a particle concentration of 12 particles per IA volume.

| Method Name   | Absolute Error (Mean) | Relative Error (Mean) |
|---|-----------------------|-----------------------|
| PIV warping   | −33.286 dBpx          | −45.327 dB            |
| PIV warping, final sub-pixel                        | −38.448 dBpx          | −50.490 dB            |
| Liu–Shen combined with Lucas–Kanade                 |                       |                       |
| Hybrid PIV with Lucas–Kanade                        | −35.738 dBpx          | −47.780 dB            |
| Hybrid PIV with Liu–Shen combined with Lucas–Kanade | −37.821 dBpx          | −49.862 dB            |

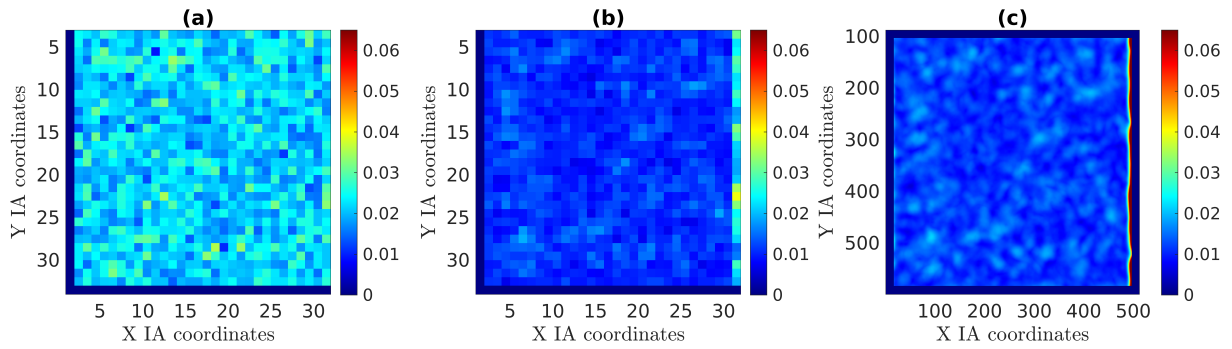


Figure 5.12: Spatial distribution of absolute errors represented through Figures **a**, **b**, or **c**. Relative errors are not shown since they are essentially negligible everywhere. These figures are for the uniform flow, with a 4.0 px maximum velocity vector magnitude, 8-bit depth images, and a particle number concentration of 12 particles per IA volume. Figures (**a–c**) have the same characteristics as their equivalents (**a1–c1**) in Figure 5.8

### 5.2.3 Laboratory Databases

#### Description of Databases and Validation Criteria

We employed three different PIV image databases from different experiments, all grayscale with 8-bit depth images, but with varying image quality with respect to particle spot sizes, concentrations, and background noise. A different flow type is represented by each database:

- **Cylinder database:** The images were obtained from a flow around a wall-mounted smooth circular cylinder. The flow is characterized by the Reynolds number based on the cylinder diameter,  $Re = dU_b/\nu$  equal to 1750, where  $d$  is the cylinder diameter,  $U_b$  is the approaching velocity, and  $\nu$  is the kinematic viscosity. The measured plane corresponds to the horizontal plane (longitudinal-spanwise direction) located at 60% of the flow depth. The database is composed of 4500 poor-quality PIV image pairs having a  $1600 \times 1200$  px<sup>2</sup> resolution. The particle spot sizes are not within the optimal PIV range, being on the smaller side, and the particle concentration is also low. There is a background pattern in the PIV images, and there is a cable harness that oscillates during the PIV recording (Figure 5.13a). This is an image quality that should also prove difficult for OpF, since OpF is known to also have difficulties with average particle diameters smaller than 2.0 px. Still, we wish to compare the performance of hybrid PIV with this image database, for which we have two validations: obtain the expected peak in the energy spectrum in the cylinder's wake and find the expected properties of undisturbed flow.
- **Boundary layer database:** A database composed of 4500 fair-quality PIV image pairs with a  $1600 \times 1200$  px<sup>2</sup> resolution, acquired from a turbulent open-channel flow over a glass bead fixed bed. The acquired data correspond to a vertical plane (longitudinal bed-normal direction) located at the channel center line. The images have a static background pattern, as shown in Figure 5.13b. The seeding quality is on the average side. This database is to be validated in accordance with the known features of boundary layer flows.
- **Plunging flow database:** The images were obtained from a complex flow in a dam break flume setup. The measured plane corresponds to a vertical plane (longitudinal bed-normal direction) located in the breach region at 0.5 cm from the channel side wall. The database is composed of 1817 PIV image pairs, with a resolution of  $1920 \times 1200$  px<sup>2</sup>, adequate particle spot sizes, and good image quality with a higher particle concentration than the other two databases. The images have no special background pattern (Figure 5.13c); however, there are two regions that tend to form large air bubbles, which can affect the measurements.

Hybrid PIV started at an IA resolution of  $128 \times 128$  with 50% overlap and went to  $16 \times 16$  with 50% overlap, employing micro-warping on the second image and applying Hongwei Guo's robust linear regression of the 1D-1D Gaussian function as the sub-pixel method. Validation was also applied to PIV at the end of each adaptive step by employing the normalized median validation, employing secondary peak replacement up to the 4th secondary peak and having a final substitution backup with linear interpolation. Hybridization with Lucas–Kanade was performed by applying a single final step of the Lucas–Kanade

OpF method, either with or without a Gaussian filter. Dense results were exported without validation applied to the final OpF step. Lucas–Kanade OpF was set up with 5 iterations and a window size of 27 px, and the Gaussian filter, when applied, had a width of 3 pixels and a standard deviation ( $\sigma$ ) of 2.0 px. Hybridization with Liu–Shen combined with Lucas–Kanade followed the same base configuration as hybrid PIV with Lucas–Kanade, but with the Gaussian filter always enabled. Liu–Shen OpF was applied after the first pass with Lucas–Kanade. Liu–Shen was set up with 60 iterations, a Lagrange multiplier value of 4, and a Gaussian filter of 3 px width and a standard deviation of  $\sigma = 0.48$  px. PIV with warping with the Hongwei sub-pixel at all adaptive steps was set up in exactly the same way, with the exception that the last step with OpF was disabled. Another PIV method under test was PIV with warping that employed Hongwei Guo’s sub-pixel for the initial adaptive steps and the Lucas–Kanade OpF sub-pixel for the last adaptive step, otherwise sharing the same setup with the PIV with warping. Classic PIV with Lucas–Kanade at all adaptive steps did not employ warping at all, and it had Hongwei Guo’s sub-pixel replaced by Lucas–Kanade OpF at all adaptive steps.

## Cylinder Database

Turbulent flows around wall-mounted smooth cylinders, which have been widely studied, are characterized by alternating vortex shedding due to flow separation on the cylinder. The normalized shedding frequency, named the Strouhal number, for flows with Reynolds numbers between 1000 and 4000 is approximately constant and equal to 0.21 [41]. The Strouhal number is defined as  $St = f \times d/U_b$ , where  $f$  is the frequency,  $d$  is the cylinder diameter, and  $U_b$  is the approaching velocity.

Figure 5.14 presents the energy spectrum at a point in the wake of the cylinder, following the Taylor hypothesis. The energy peak corresponding to the shedding frequency is well identified by both PIV and the hybrid methods. The hybrid PIV method with Lucas–Kanade generates a noisier spectrum than the other methods; nevertheless, an energy peak is detectable at the expected shedding frequency.

Sufficiently outside of the cylinder influence, the theoretical condition of homogeneous and isotropic turbulence is applicable, and, thus, an inertial range of scales characterized by the  $-5/3$  law is expected [42] (pp. 248–274). The inertial range of scales corresponds to scales that are not directly affected by the energy maintenance and dissipation mechanisms and where merely a transfer of energy, from production processes to dissipation processes, occurs. For that scale’s range, the energy spectrum,  $E(k)$ , is described by  $E(k) = c\epsilon^{2/3}k^{-5/3}$ , where  $c$  is a universal constant,  $\epsilon$  is the rate of energy dissipation, and  $k$  is the wave number.

To assess the ability of the hybrid methods to improve the spatial resolution for the cylinder database, a longitudinal strip of data was selected on the left-hand-side end and employed to compute the power spectral density (PSD) functions depicted in Figure 5.15. The results for PIV methods, without an appreciable difference between methods, show, approximately, an energy spectrum with the expected  $-5/3$  slope for  $k$  between 60 and 600  $\text{m}^{-1}$ , despite its noisy shape. Likely due to the low quality of the images in this database, the smallest scales of the PIV methods deviate from the  $-5/3$  law without any physical justification. Relative to the hybrid method’s behavior, the spectra are similar to those obtained by PIV methods until  $k = 300 \text{ m}^{-1}$ , deviating from the  $-5/3$  law for larger wave numbers. The hybrid meth-

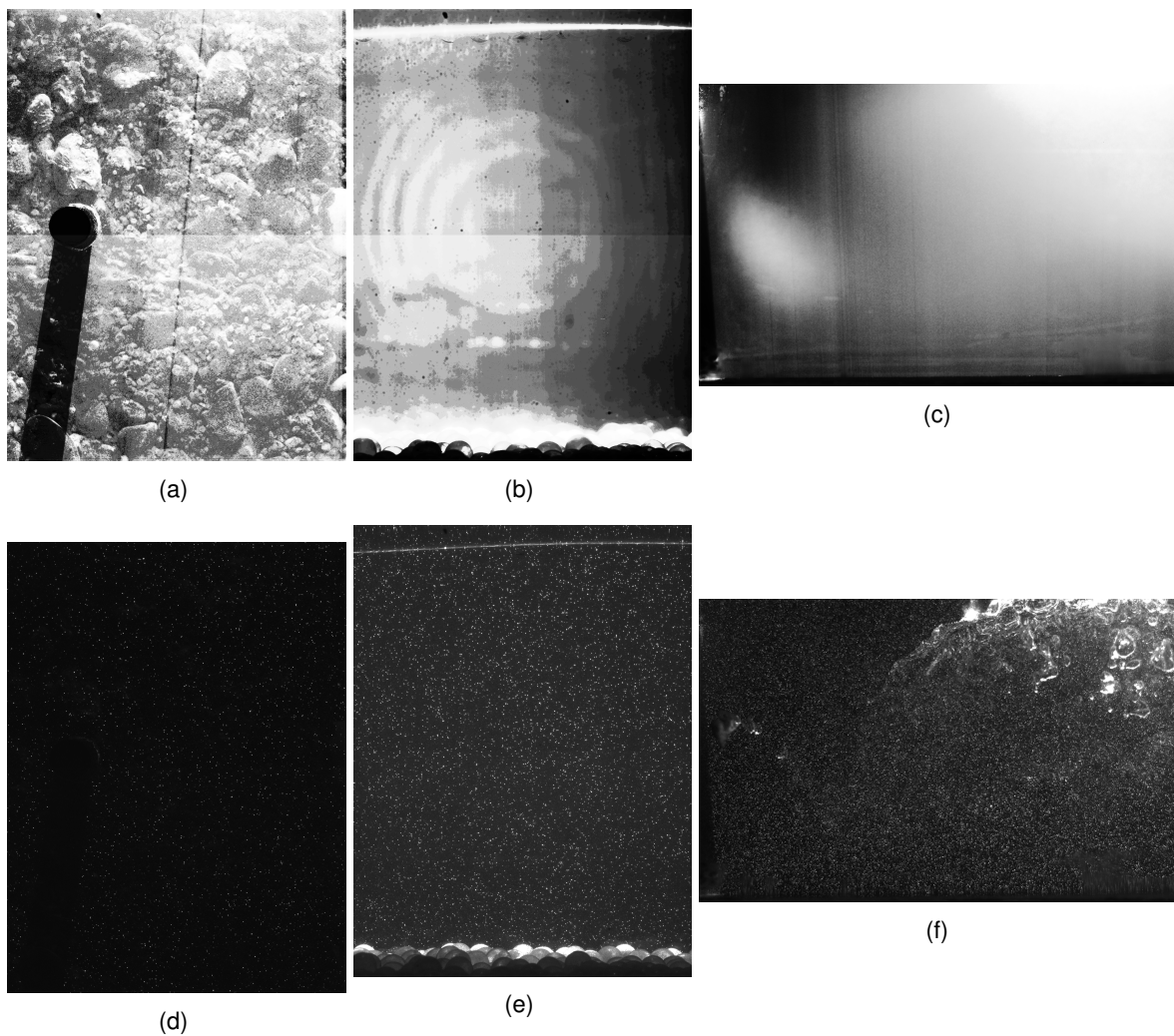


Figure 5.13: The top row shows median background images after intensity normalization. The bottom row shows sample PIV images from the image database. The flow is from left to right hand sides. **(a)** Cylinder database (background); **(b)** Boundary layer db (background); **(c)** Plunging flow database (background); **(d)** Cylinder database (PIV image); **(e)** Boundary layer db (PIV image); **(f)** Plunging flow database (PIV image).

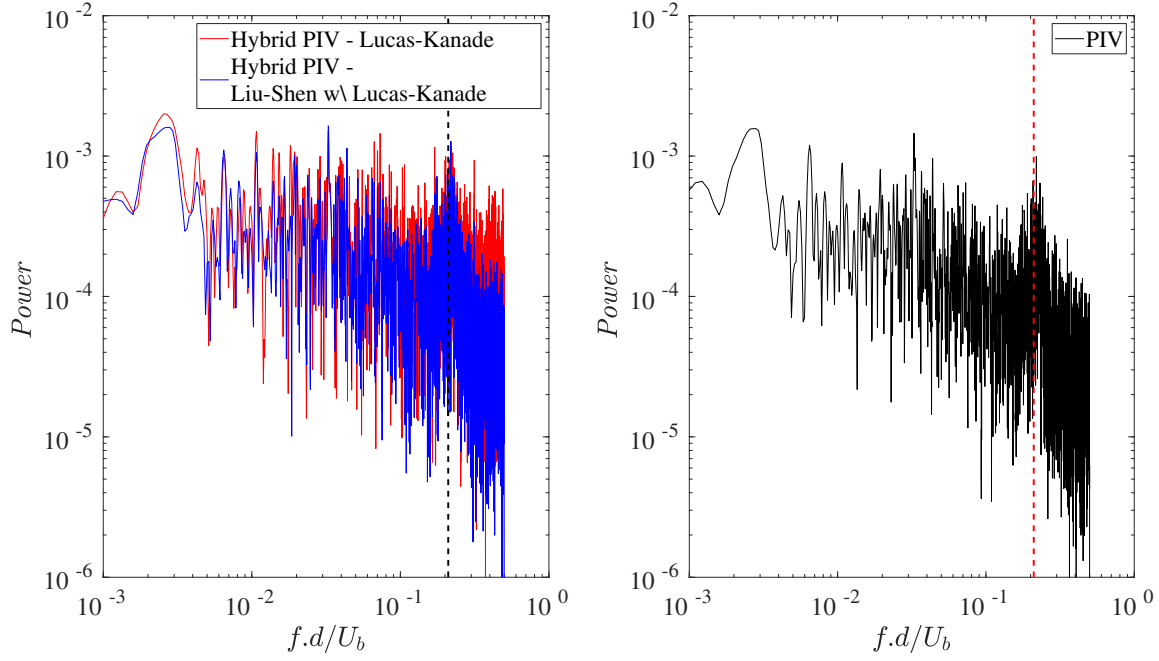


Figure 5.14: The power spectral density (PSD) in the cylinder wake as a function of the non-dimensional frequency at the normalized location ( $X = 0.58d, Y = 2.28d$ ), where  $d$  is the cylinder diameter, and the referential origin is the tip point on the lee side of the cylinder. The vertical dashed lines represent the Strouhal number of 0.21.

ods with Lucas–Kanade only, with or without a Gaussian filter, show a slope smaller than  $-5/3$ , while the method of Liu–Shen with Lucas–Kanade presents a faster rate of energy transfer to smaller scales. None of the methods recovers the expected  $-5/3$  slope, indicating that, for this database of poor quality, the hybrid methods are not able to adequately improve the spatial resolution of the spatial data.

### Boundary Layer Database

A turbulent open-channel flow over a rough fixed bed is characterized by the well-known boundary layer theory. To evaluate the adequacy of employing hybrid methods to extend the spatial resolution, we computed, for each tested method, the auto-correlation function and the energy spectrum of a longitudinal profile of the flow velocity sampled at 7% of the flow depth (i.e., a line parallel to the bed at about 5 mm above the crest level). The velocity profile consists of a series of time-averaged longitudinal velocity components.

The auto-correlation presented in Figure 5.16 shows no appreciable difference among different PIV and hybrid methods. Moreover, for all the tested methods, the auto-correlation function is approximately zero for a space lag  $r = 1$  cm. The main difference is observed for the smallest space lags. Hybrid PIV with dense Lucas–Kanade without a Gaussian filter and classic PIV with the Lucas–Kanade sub-pixel show larger noise levels than the other methods.

Figure 5.17 presents the power spectrum density (PSD) data, showing the expected  $-5/3$  slope for all methods except for the hybrid PIV with Liu–Shen combined with Lucas–Kanade. Nevertheless, hybrid PIV with LK and classic PIV with LK for all steps led to better results, as the  $-5/3$  slope applies to a larger range of scales. The inertial range of scales of the studied boundary layer flow is expected to

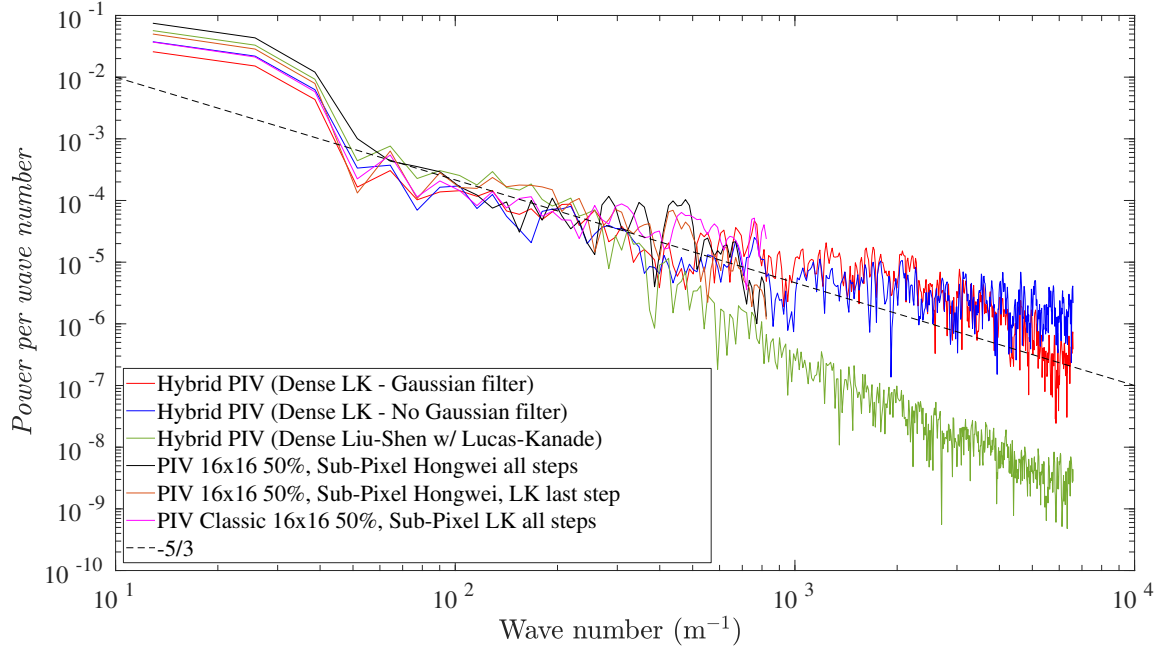


Figure 5.15: Cylinder database—the power spectral density (PSD) function for a spatial series of longitudinal velocity components at  $y = 7.626d$ , the region far from the cylinder wake. The center of the cylinder is located at  $x = 1.540d$ ,  $y = 5.322d$ .

span to scales on the order of a few millimeters. The PIV methods with Hongwei generate PSD functions deviating from the  $-5/3$  slope close to the tail, likely due to noise effects, rendering an unexpected short inertial range of scales. The results from the hybrid methods with dense LK are encouraging, as the spectral analysis renders the extension of the inertial range of scales as expected. An increase in the noise level is visible for the smallest scales, in accordance with the observations from the auto-correlation function; however, this noise level does not compromise the main result: the identification of the inertial range of scales spanning from  $k = 200 \text{ m}^{-1}$  to  $k = 5000 \text{ m}^{-1}$ .

In summary, the results for images of moderately good quality indicate that the hybrid methods with dense LK can improve the spatial resolution relative to classic PIV methods. Such an improvement is of paramount importance for turbulence analysis, allowing, for example, the computation of Taylor's micro-scale from the auto-correlation function or the dissipation rate from the energy spectrum.



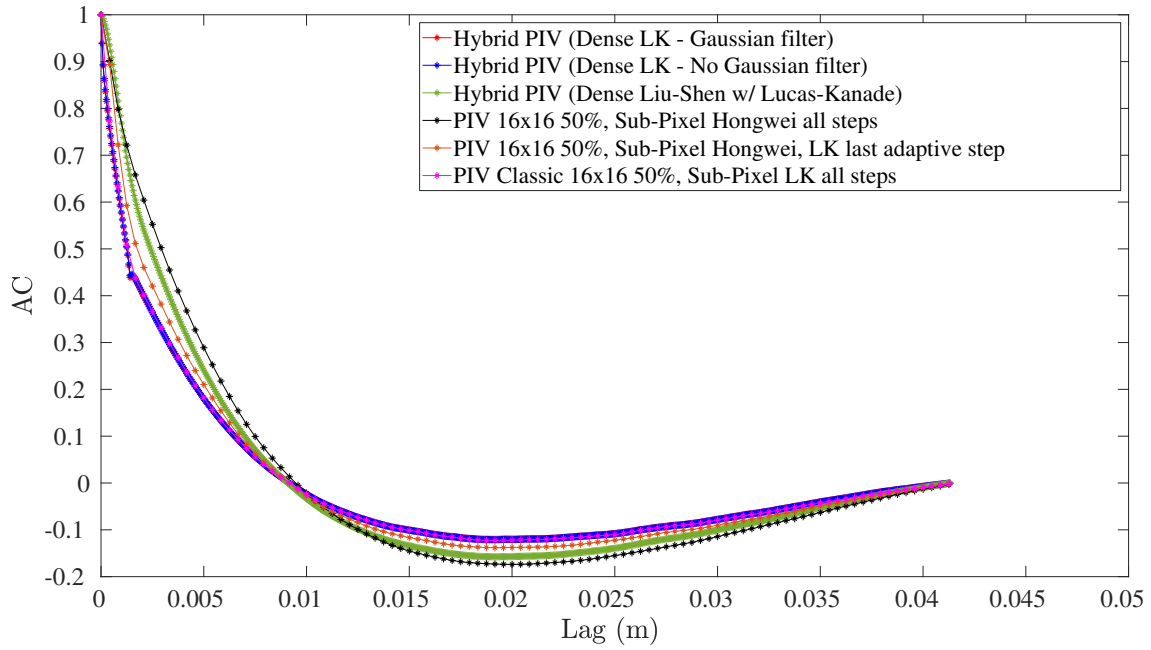


Figure 5.16: The boundary layer database—the auto-correlation function of the spatial series of longitudinal velocity components at at 7% of the flow depth.

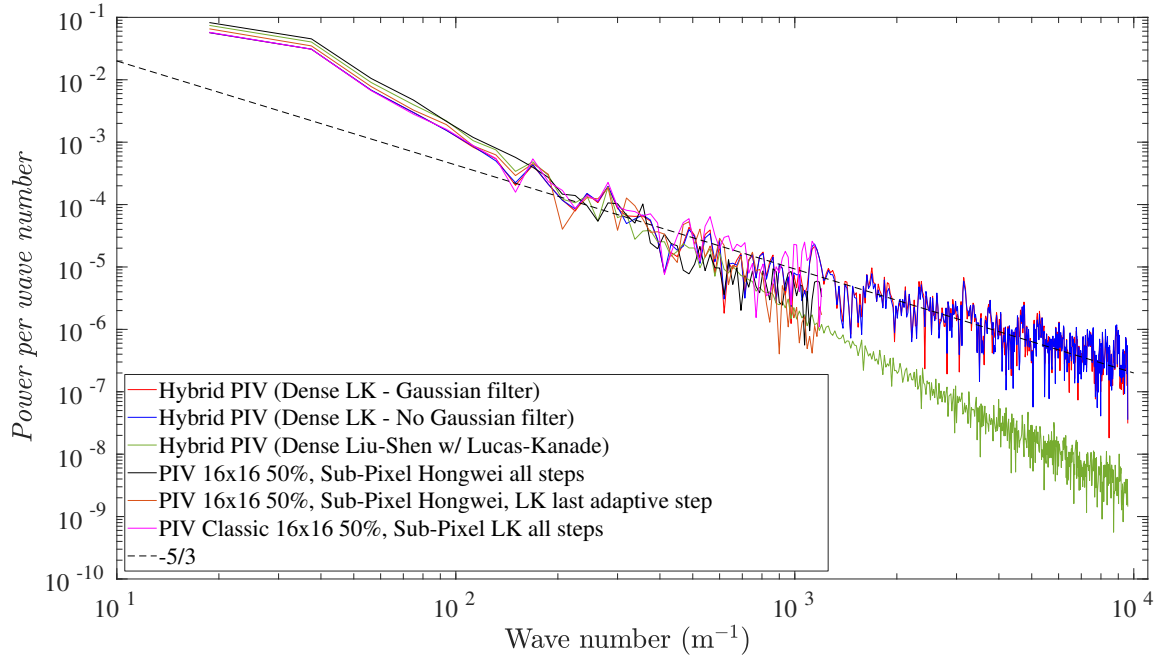


Figure 5.17: The boundary layer database—the power spectral density (PSD) function of the spatial series of longitudinal velocity components at at 7% of the flow depth.

## **Plunging Flow Database**

In this database, all the methods have smooth and compatible auto-correlation functions (see Figure 5.18). With respect to the power spectral density (PSD), as shown in Figure 5.19, all methods present high compatibility, with the hybrid PIV methods achieving the full pixel resolution for this database. The Gaussian filter applied to the hybrid PIV Lucas–Kanade results in a power reduction for the final range of wave numbers, thus possibly reducing image noise to some extent but also potentially modifying the spectral information. Hybrid PIV with Liu–Shen combined with Lucas–Kanade again extends the PIV slope to higher wave numbers, essentially following the PIV slope with respect to the PSD.

The results from this database, composed of images of good quality, lead to the conclusion that the hybrid methods can be adequately employed to improve the spatial resolution relative to classic PIV methods.

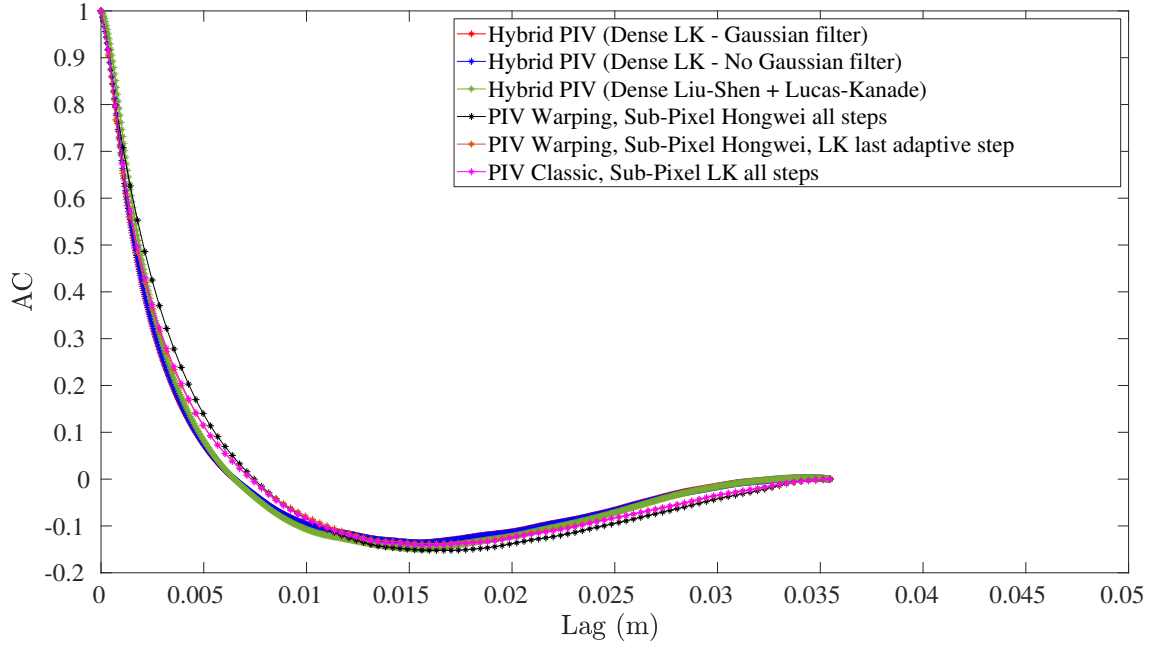


Figure 5.18: The plunging flow database—the auto-correlation function for the bed-normal velocity component for a region with similar mean velocities.

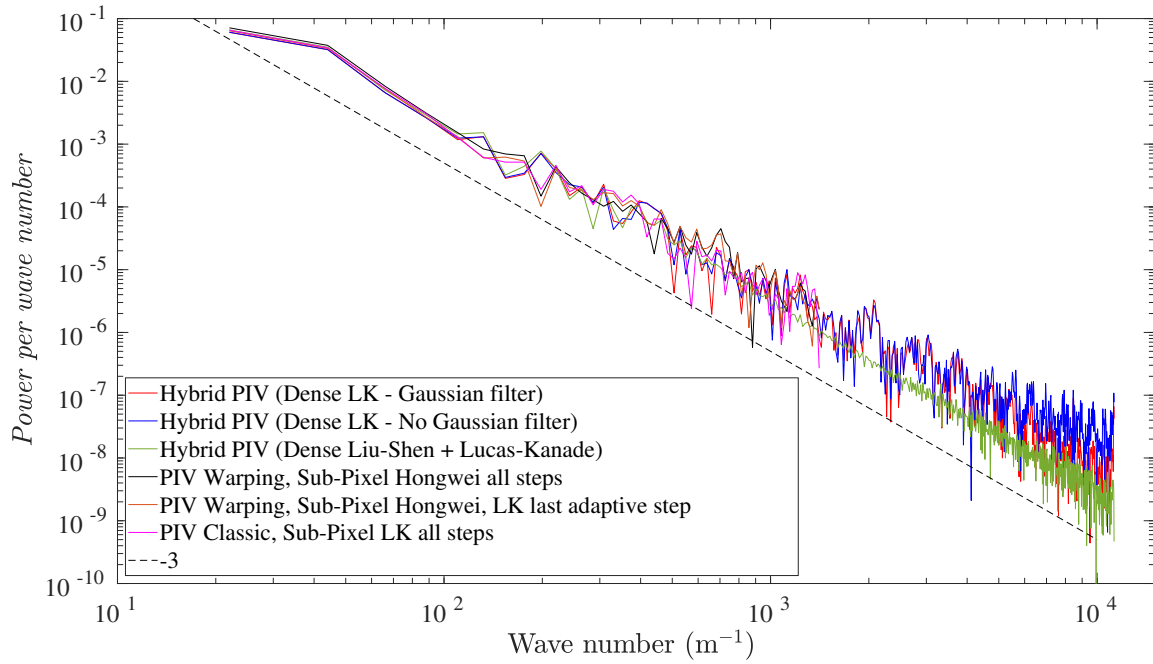


Figure 5.19: The plunging flow database—the power spectral density (PSD) function for the bed-normal velocity component in a region with similar mean velocities.

## 5.3 Methods and Materials

We cannot provide the test databases online due to their sheer size, as we would not be able to guarantee the medium- to long-term availability of the contents. We thus prefer to provide instructions to generate similar ones for the synthetic image database. As for the experimental databases, the authors are willing to provide them upon request.

It should, however, be noted that these databases are not a critical part of the article and that their role is just to document the PIV and hybrid PIV methods' capabilities. As such, the results presented are expected to be compatible with other results obtained from other databases with similar characteristics.

Other than that, as of this writing, the recommended version to reproduce these results is QuickLabPIV-ng v0.8.7, obtainable from <https://github.com/CoreRasurae/QuickLabPIV-ng/releases/tag/v0.8.7> (accessed on 29 March 2024). The software itself does most of the computation without requiring external libraries/frameworks; namely, image filtering, cross-correlation, and Lucas–Kanade and Liu–Shen optical flow methods are implemented internally, which should protect the software from bit-rot, as well as unexpected behavioral changes coming from external library updates. Even with this effort made, it was not completely possible to avoid an important dependency on the Aparapi (<https://aparapi.com> (accessed on 29 March 2024), <https://git.cleverthis.com/cleverthis/aparapi/aparapi/-/releases/v3.0.2> (accessed on 29 March 2024)) library/framework, which enables High-Performance Computing (HPC) for Java in systems with GPU or CPU devices supporting OpenCL.

Besides the software, it requires a computer with at least 32 to 64 GB of RAM for medium- to large-sized databases. Ideally, the system should be a multi-core computer with at least one AMD® or NVIDIA® GPU. Discrete Intel® GPUs have not been tested yet, but integrated Intel® GPUs are known to produce invalid results with the current versions of QuickLabPIV-ng/Aparapi. A Java virtual machine compatible with at least Java 8 should be employed, and it has been tested in both Windows® and Linux environments. Currently, QuickLabPIV-ng only supports image depths of 8 bits from the GUI. We ran our databases on a system with 64GB of RAM by invoking QuickLabPIV-ng with `java -Xmx48G -jar QuickLabPIVng.jar`.

A synthetic image database can be easily generated from piv-image-generator [37] but using the latest code from <https://git.qoto.org/CoreRasurae/piv-image-generator/-/releases/v1.2.0> (accessed on 29 March 2024). All that is required is to edit the main configuration/script file 'exampleAllTestImagesMain.m' and adjust the following contents:

```
displayFlowField=false;
closeFlowField=true;

flows={'uniform' 'parabolic' 'stagnation' ...
'rankine_vortex' 'rk_uniform'};
bitDepths=[8];
deltaXFactor=[0.05 0.10 0.25];
```

```

particleRadius=[0.5 1.0 1.5 3.0];
Ni=[1 6 12 16];
noiseLevel=[0 5 15];
outOfPlaneStdDeviation=[0.025 0.050 0.100];
numberOfRuns=10;

generatePIVImagesWithAllParametersCombinations;

```

All software was developed in-house and is available as open-source. The PIV hardware equipment, namely the PIV Laser lighting and image acquisition hardware, was obtained from Dantec Dynamics A/S, Skovlunde, Denmark.

## 5.4 Conclusions and Recommendations for Further Analysis

The main conclusion of this research is that hybrid PIV-OpF can improve PIV's resolution and ability to provide data adequate to analyze fine-scale turbulence. While a better resolution of the mean flow is easy to achieve, turbulence statistics are improved only if the raw PIV images are of good quality, i.e., are within the optical range of tracer density and spot size, and have low optical noise. Hybrid PIV with Liu–Shen and Lucas–Kanade offers the highest accuracy in the description of the mean flow. However, in real databases, it may not closely follow the expected PSD slope of  $-5/3$ , despite extending the slope for higher frequencies, suggesting a potential for noise filtering.

To analyze turbulence, the Lucas–Kanade algorithm alone is preferable. It correctly reproduces the flow's macro-scales and the slope of the power spectral density function and does so in locally isotropic homogeneous turbulence. It is also computationally more efficient.

Integrating Liu–Shen with Lucas–Kanade optical flow (OpF) as an alternative to interpolation peak reconstitution with sub-pixel accuracy is a valid alternative, especially in essentially 2D flows with high-quality images. In this case, the method achieves the accuracy of traditional PIV methods employing warping. PIV's last adaptive step with a Lucas–Kanade OpF step maintains the standard PIV workflow while leveraging the precision benefits of optical flow techniques.

From our experience, independently of the method, the optimal image conditions include particle sizes around 3 px and at least 6 particles per IA, improving with higher concentrations. For the analysis of PIV sequences, a tailored approach is recommended for optimal results. For classic PIV incorporating optical flow (OpF) for sub-pixel refinement, the sparse Lucas–Kanade method is preferred. If initial sub-pixel interpolation deviates significantly from expectations, it is advisable to limit the OpF application to the final refinement stage. For advanced PIV techniques, including window warping, a combination of Hongwei Guo's robust Gaussian regression and dense Lucas–Kanade with Liu-Shen OpF, adjusted for the last adaptive step, enhances accuracy. The Lagrange multiplier in the Liu–Shen method is sensitive to the pixel's brightness range, so it is recommended to normalize the pixel intensities to a fixed range, which QuickLabPIV-ng does automatically. For validation with vector substitution, it is recommended to

use a normalized median combined with multi-peak replacement with four peaks around the main peak and a search kernel set with a width of 3 pixels. These strategies, validated through synthetic imagery, align with the evolving needs of PIV analysis, ensuring precise and reliable measurements under various experimental conditions.

Our preliminary examination across the three PIV image databases with real experimental data suggests promising potential for hybrid PIV techniques, particularly when integrating Lucas–Kanade methods, in broadening the scope of turbulence measurements achievable with PIV. However, this exploration is just the beginning. More comprehensive studies are imperative to fully harness and understand the capabilities and limitations of these advanced methodologies. Additionally, the impact of Gaussian filtering on the fidelity of Taylor micro-scale readings warrants further investigation, as it could inadvertently obscure crucial data.

## 5.5 References

- [1] Scarano, F. Tomographic PIV: Principles and practice. *Meas. Sci. Technol.* **2012**, *24*, 012001. <https://doi.org/10.1088/0957-0233/24/1/012001>.
- [2] Aleixo, R.; Soares-Frazão, S.; Zech, Y. Velocity-field measurements in a dam-break flow using a PTV Voronoï imaging technique. *Exp. Fluids* **2011**, *50*, 1633–1649.
- [3] Capart, H.; Young, D.; Zech, Y. Voronoï imaging methods for the measurement of granular flows. *Exp. Fluids* **2002**, *32*, 121–135.
- [4] Horn, B.K.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203. [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2).
- [5] Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI’81, Vancouver, BC, Canada, 24–28 August 1981; Volume 2, pp. 674–679.
- [6] Aubert, G.; Deriche, R.; Kornprobst, P. Computing Optical Flow via Variational Techniques. *SIAM J. Appl. Math.* **1999**, *60*, 156–182. <https://doi.org/10.1137/S0036139998340170>.
- [7] Farnebäck, G. Two-Frame Motion Estimation Based on Polynomial Expansion. In *Image Analysis*; Bigun, J., Gustavsson, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 363–370. [https://doi.org/10.1007/3-540-45103-X\\_50](https://doi.org/10.1007/3-540-45103-X_50).
- [8] Tomasi, C.; Kanade, T. *Detection and Tracking of Point Features*; Technical Report CMU-CS-91–132, Carnegie Mellon University ; 1991; [https://www.ri.cmu.edu/pub\\_files/pub2/tomasi\\_c\\_1991\\_1/tomasi\\_c\\_1991\\_1.pdf](https://www.ri.cmu.edu/pub_files/pub2/tomasi_c_1991_1/tomasi_c_1991_1.pdf).
- [9] Anandan, P. A computational framework and an algorithm for the measurement of visual motion. *Int. J. Comput. Vis.* **1989**, *2*, 283–310. <https://doi.org/10.1007/BF00158167>.

- [10] Wills, J.; Agarwal, S.; Belongie, S. A Feature-based Approach for Dense Segmentation and Estimation of Large Disparity Motion. Int. J. Comput. Vis. **2006**, 68, 125–143. <https://doi.org/10.1007/s11263-006-6660-3>.
- [11] Ohta, Y.; Kanade, T. Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. IEEE Trans. Pattern Anal. Mach. Intell. **1985**, PAMI-7, 139–154. <https://doi.org/10.1109/TPAMI.1985.4767639>.
- [12] Adam, P.; Burg, B.; Zavidovique, B. Dynamic programming for region based pattern recognition. In Proceedings of the ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing, Tokyo, Japan, 7–11 April 1986; Volume 11, pp. 2075–2078. <https://doi.org/10.1109/ICASSP.1986.1168952>.
- [13] Heeger, D.J. Optical flow using spatiotemporal filters. Int. J. Comput. Vis. **1988**, 1, 279–302. <https://doi.org/10.1007/BF00133568>.
- [14] Fleet, D.J.; Jepson, A.D. Computation of component image velocity from local phase information. Int. J. Comput. Vis. **1990**, 5, 77–104. <https://doi.org/10.1007/BF00056772>.
- [15] Dosovitskiy, A.; Fischer, P.; Ilg, E.; Häusser, P.; Hazirbas, C.; Golkov, V.; Smagt, P.v.d.; Cremers, D.; Brox, T. FlowNet: Learning Optical Flow with Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2758–2766. <https://doi.org/10.1109/ICCV.2015.316>.
- [16] Ren, Z.; Yan, J.; Ni, B.; Liu, B.; Yang, X.; Zha, H. Unsupervised Deep Learning for Optical Flow Estimation. In Proceedings of the AAAI Conference on Artificial Intelligence San Francisco, CA, USA, 4–9 February 2017; Volume 31.
- [17] Quénot, G.; Pakleza, J.; Kowalewski, T.A. Particle image velocimetry with optical flow. Exp. Fluids **1998**, 25, 177–189. <https://doi.org/10.1007/S003480050222>.
- [18] Ruhnau, P.; Kohlberger, T.; Schnörr, C.; Nobach, H. Variational optical flow estimation for particle image velocimetry. Exp. Fluids **2005**, 38, 21–32. <https://doi.org/10.1007/s00348-004-0880-5>.
- [19] Corpetti, T.; Heitz, D.; Arroyo, G.; Mémin, E.; Santa-Cruz, A. Fluid experimental flow estimation based on an optical-flow scheme. Exp. Fluids **2006**, 40, 80–97. <https://doi.org/10.1007/s00348-005-0048-y>.
- [20] Ruhnau, P.; Schnörr, C. Optical Stokes flow estimation: an imaging-based control approach. Exp. Fluids **2007**, 42, 61–78. <https://doi.org/10.1007/s00348-006-0220-z>.
- [21] Heitz, D.; Héas, P.; Mémin, E.; Carlier, J. Dynamic consistent correlation-variational approach for robust optical flow estimation. Exp. Fluids **2008**, 45, 595–608. <https://doi.org/10.1007/s00348-008-0567-4>.

- [22] Heitz, D.; Mémin, E.; Schnörr, C. Variational fluid flow measurements from image sequences: synopsis and perspectives. *Exp. Fluids* **2010**, *48*, 369–393. <https://doi.org/10.1007/s00348-009-0778-3>.
- [23] Le Besnerais, G.; Champagnat, F. Dense optical flow by iterative local window registration. In Proceedings of the IEEE International Conference on Image Processing 2005, Genoa, Italy, 11–14 September 2005; Volume 1, pp. I–137. <https://doi.org/10.1109/ICIP.2005.1529706>.
- [24] Champagnat, F.; Plyer, A.; Le Besnerais, G.; Leclaire, B.; Davoust, S.; Le Sant, Y. Fast and accurate PIV computation using highly parallel iterative correlation maximization. *Exp. Fluids* **2011**, *50*, 1169–1182. <https://doi.org/10.1007/s00348-011-1054-x>.
- [25] Liu, T.; Shen, L. Fluid flow and optical flow. *J. Fluid Mech.* **2008**, *614*, 253–291. <https://doi.org/10.1017/S0022112008003273>.
- [26] Schmidt, B.E.; Sutton, J.A. High-resolution velocimetry from tracer particle fields using a wavelet-based optical flow method. *Exp. Fluids* **2019**, *60*, 37. <https://doi.org/10.1007/s00348-019-2685-6>.
- [27] Page, W.E.; Schmidt, B.E.; Sutton, J.A., Experimental Assessment of Wavelet-based Optical Flow Velocimetry (wOFV) as Applied to Tracer Particle Images from Free Shear Flows. In *AIAA Scitech 2020 Forum*; American Institute of Aeronautics and Astronautics, Inc.: Reston, USA 2020. <https://doi.org/10.2514/6.2020-1022>.
- [28] Yang, Z.; Johnson, M. Hybrid particle image velocimetry with the combination of cross-correlation and optical flow method. *J. Vis.* **2017**, *20*, 625–638. <https://doi.org/10.1007/s12650-017-0417-7>.
- [29] Głomb, G.; Świrniak, G. A hybrid method for velocity field of fluid flow estimation based on optical flow. In *Optical Measurement Systems for Industrial Inspection XI*; Lehmann, P., Osten, W., Júnior, A.A.G., Eds.; International Society for Optics and Photonics, SPIE: Washington, USA 2019; Volume 11056, pp. 969–980. <https://doi.org/10.1117/12.2525711>.
- [30] Seong, J.H.; Song, M.S.; Nunez, D.; Manera, A.; Kim, E.S. Velocity refinement of PIV using global optical flow. *Exp. Fluids* **2019**, *60*, 174. <https://doi.org/10.1007/s00348-019-2820-4>.
- [31] Brox, T.; Bruhn, A.; Papenberg, N.; Weickert, J. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In Proceedings of the Computer Vision—ECCV 2004, Prague, Czech Republic, 11–14 May 2004; Pajdla, T., Matas, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 25–36.
- [32] Liu, T.; Salazar, D.M.; Fagehi, H.; Ghazwani, H.; Montefort, J.; Merati, P. Hybrid Optical-Flow-Cross-Correlation Method for Particle Image Velocimetry. *J. Fluids Eng.* **2020**, *142*, 054501, <https://doi.org/10.1115/1.4045572>.



- [33] Liu, T.; Salazar, D. OpenOpticalFlow\_PIV: An Open Source Program Integrating Optical Flow Method with Cross-Correlation Method for Particle Image Velocimetry. J. Open Res. Softw. **2021**, 9, 3. <https://doi.org/10.5334/jors.326>.
- [34] Ouyang, Z.; Yang, H.; Huang, Y.; Zhang, Q.; Yin, Z. A circulant-matrix-based hybrid optical flow method for PIV measurement with large displacement. Exp. Fluids **2021**, 62, 233. <https://doi.org/10.1007/s00348-021-03317-1>.
- [35] Mendes, L.P.N.; Ricardo, A.M.C.; Bernardino, A.J.M.; Ferreira, R.M.L. A comparative study of optical flow methods for fluid mechanics. Exp. Fluids **2021**, 63, 7. <https://doi.org/10.1007/s00348-021-03357-7>.
- [36] Ricardo, A.M.; Koll, K.; Franca, M.J.; Schleiss, A.J.; Ferreira, R.M.L. The terms of turbulent kinetic energy budget within random arrays of emergent cylinders. Water Resour. Res. **2014**, 50, 4131–4148. <https://doi.org/https://doi.org/10.1002/2013WR014596>.
- [37] Mendes, L.; Bernardino, A.; Ferreira, R.M. piv-image-generator: An image generating software package for planar PIV and Optical Flow benchmarking. SoftwareX **2020**, 12, 100537. <https://doi.org/https://doi.org/10.1016/j.softx.2020.100537>.
- [38] Raffel, M.; Willert, C.; Wereley, S.; Kompenhans, J. Particle Image Velocimetry—A Practical Guide; Springer: Berlin/Heidelberg, Germany, 2007. <https://doi.org/10.1007/978-3-540-72308-0>.
- [39] Guo, H., A Simple Algorithm for Fitting a Gaussian Function. In Streamlining Digital Signal Processing; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2012; Chapter 31, pp. 297–305. <https://doi.org/https://doi.org/10.1002/9781118316948.ch31>.
- [40] Mendes, L.; Bernardino, A.; Ferreira, R. Synthetic PIV Image Generator with Ground-Truth. 2020. Available online: <https://git.qoto.org/CoreRasurae/piv-image-generator> (accessed on 29 November 2024).
- [41] Norberg, C. Fluctuating lift on a circular cylinder: Review and new measurements. J. Fluids Struct. **2003**, 17, 57–96. [https://doi.org/https://doi.org/10.1016/S0889-9746\(02\)00099-3](https://doi.org/https://doi.org/10.1016/S0889-9746(02)00099-3).
- [42] Tennekes, H.; Lumley, J.L. A First Course in Turbulence; MIT Press: Cambridge, MA, USA, 1972.



## Chapter 6

# Conclusions

This thesis has studied, with the help of the developed open-source piv-image-generator tool, the application of optical flow methods to velocimetry in fluid mechanics applications. Novel optical flow (OpF) combinations were analyzed and benchmarked, namely the novel Liu-Shen combined with Lucas-Kanade and the novel Liu-Shen combined with Farnebäck. Both the Farnebäck method and the dense Lucas-Kanade OpF method variant from the computer vision field were revisited, the latter being mostly neglected for fluid mechanics applications. It was found that the novel OpF methods combinations present increased accuracy for the tested synthetic image database and are compatible with the desired image and particle spot characteristics required by the PIV, fact that led to a novel hybrid PIV-OpF method with the Liu-Shen combined with Lucas-Kanade OpF. A novel open-source HPC software, QuickLabPIV-ng, which implements the said hybrid PIV-OpF method, was presented and made available to the public. QuickLabPIV-ng was used to process real PIV image databases of turbulent flows composed of up to 4500 image pairs. The novel hybrid PIV-OpF method was evaluated in terms of its performance with turbulent flows, considering both the Power Spectral Density (PSD) and autocorrelation functions. It was found that the hybrid PIV-OpF with the dense Lucas-Kanade method is suitable for turbulent flows and is able to extend the turbulence range down to much lower scales than the standard PIV, provided that the PIV image databases are of good quality. It was also found that the hybrid PIV-OpF with Liu-Shen combined with Lucas-Kanade is able to increase the accuracy of the mean velocities estimates, but fails to preserve the characteristics of turbulence at small scales. It was found that the most difficult flow considered in the synthetic benchmark database for the hybrid and OpF methods is the Poiseuille flow. The optimal PIV and OpF conditions encompass particles spot sizes of 2 px and at least 6 particles per IA, but improving with higher concentrations. A low-cost portable PIV LED lighting unit was also engineered and validated for PIV image acquisition. The Liu-Shen combined with Lucas-Kanade and Liu-Shen combined with Farnebäck OpF methods are lightweight methods when compared to many of the OpF methods currently employed for fluid mechanics applications, because they require less math operations and are able to converge in a small number of iterations. In fact 5 iterations were sufficient for Lucas-Kanade and Farnebäck methods to achieve the best accuracy and Liu-Shen is able to converge in 60 iterations, which results in an improvement over the Physics based optical flow since,

Horn-Schunck typically requires hundreds of iterations, up to 500 iterations and the Liu-Shen iterations are the same 60 iterations, thus the Physics based Optical flow requires much more operations to complete. This way the Liu-Shen combined with Lucas-Kanade should be a good candidate to test in edge computing devices.

## 6.1 Achievements

An open-source software for generating synthetic PIV images with customizable flow types has been released to the public, under the name `piv-image-generator`, and accompanied with a journal publication [1] and the source code has been open-sourced (<https://git.qoto.org/CoreRasurae/piv-image-generator>). This software has been adopted by several users, namely researchers and PhD students from several faculties around the world.

With the `piv-image-generator` tool and custom Python implementations of original code adapted from OpenCV, or from open-source code in MATLAB it was possible to create a reference implementation of OpF methods and use them to benchmark several OpF methods, namely: the novel Liu-Shen combined with Lucas-Kanade; the novel Liu-Shen combined with Farnebäck; Physics based optical flow; the Farnebäck; the dense Lucas-Kanade and the Horn-Schunck methods to make a comparative study of OpF methods that resulted in a journal article in Experiments in Fluids [2]. That study also looked into the difficult scenarios posed by some flow types, especially the planar Poiseuille flow. This study revealed that Farnebäck and dense Lucas-Kanade could be combined with the Liu-Shen method increasing the overall accuracy and that the dense Lucas-Kanade also worked well at 8 bit-depth, which Farnebäck could not handle so well. Overall it was found that Lucas-Kanade with Liu-Shen was a performant method and that OpF was a real contender as a PIV substitute. The considered OpF methods have shown compatibility with the required image PIV characteristics, suggesting that the methods could be combined together. The open-source OpF reference implementation of the considered OpF methods was made available in <https://github.com/CoreRasurae/OpticalFlow-RI>.

Further work led to the release of a highly-performant and novel Hybrid PIV solution named QuickLabPIVng providing a novel hybrid PIV-OpF method with the Liu-Shen combined with dense Lucas-Kanade. QuickLabPIV-ng can also be used to test new PIV processing methods and is extensible due to its Object Oriented software and Software Patterns based design. It is a HPC software with multi-threading CPU support and GpGPU massive parallelization support with Aparapi and OpenCL. It tries to be more user friendly with its GUI, being fully configurable by that interface and exports the results in MATLAB compatible file format. This software has proven to be useful and is now being utilized by several students, including: one PhD student at Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, co-supervised by researchers at IST, one PhD student at IST, the students of the Joint initiative EMI - FM committees of IAHR (Jan-Mar 2024), students of the 8th W.A.T.E.R. Summer School, Strasbourg, France. The software was published as open-source in Github (<https://github.com/CoreRasurae/QuickLabPIV-ng>) to disseminate the adoption of PIV even at early stages, allowing undergraduate students to make contact with this technology early on, and so

that it can be improved by other researchers.

With QuickLabPIV-ng it was possible to process large PIV databases with hybrid PIV methods and assess the performance of hybrid PIV with dense Lucas-Kanade and with Liu-Shen combined with Lucas-Kanade. Those methods were studied with respect to real PIV databases with several image qualities obtained in experimental setups with recirculating flumes in a laboratory. It was possible to show that hybrid PIV can improve the spatial resolution of good quality PIV image databases with respect to turbulent scales recovery, but for lower quality PIV image databases, no gains are obtained with respect to normal PIV. It was also found that the hybrid PIV with dense Lucas-Kanade seems to be the most suited for turbulent flows, while Liu-Shen combined with Lucas-Kanade is better suited at improving the mean velocity estimates, while failing to preserve the turbulence characteristics at small scales. It can also be said that applying Liu-Shen combined with Lucas-Kanade always improves the mean velocity estimates even for low quality PIV image databases. The study also found that the critical flow points for all the methods are the spots with acceleration discontinuities, like walls, or zero velocity regions, or boundaries between different flow regimes. The results were published in the Water journal [3] as an open-access publication.

A portable low-cost LED PIV lighting unit was also designed, built and tested and has proven to be usable for teaching PIV with undergraduate students, or for field PIV measurements. It has been utilized by a PhD student at Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, co-supervised by researchers at IST.

## 6.2 Future Work

It was not possible to achieve the real-time behavior since it would have been a significant development effort that could easily de-rail. It would not be a wise approach to start directly implementing OpF methods on hardware without having confidence in their results first. A safer approach of validating the base algorithms and having a near real time was preferred, or at least one performant and working implementation on software running on a GpGPU massively parallel accelerated workstation. Future work would involve testing the method in a pure pyramidal OpF way and seeing how the accuracy and computation times compare with respect to the hybrid PIV solution. Only then proceeding to a pure OpF implementation on a FPGA to achieve embedded real-time performance, after which it should be optimized for low power consumption and high throughput.

Further validation should be performed regarding Liu-Shen combined with Lucas-Kanade method and with dense Lucas-Kanade method in respect to turbulence characteristics recovery against real databases, enlarging the test cases for higher statistical confidence over those methods capabilities.



## Appendix A

# Portable PIV LED lighting unit and experimental test

This small chapter gives an overview of the portable PIV LED lighting unit, from the unit itself to its experimentation in the laboratory and to final PIV image processing with QuickLabPIV-ng.

### A.1 Characteristics

A portable cost-effective LED PIV unit was engineered, built and tested from the ground up, as depicted in Figure A.1. It was designed to be cheaper and more flexible than the current LED units available in the market as far as I know. It has a rechargeable lead acid battery that allows 1h of continuous operation allowing its use in the field and includes a diagnostics and status color LCD screen, see Figure A.2 and is remotely controlled by radio frequency by means of a digital communications protocol. The green LED is 65W or about 2000 to 2300 lm at 13.5A of continuous light, despite higher pulsed mode being supported by the hardware, that mode of operation was not implemented in the firmware. The light is dimmable with 4096 power levels. All the hardware, printed circuit board, firmware and enclosure were built from scratch taking about 4 months at full-time to be completed.

The hardware also includes a fiber temperature monitor assembly that couples to the fiber bundle metal structure at its junction with the LED to protect from damaging the fiber due to overtemperature exposure.

A color LCD screen is also part of the unit and provides information about the status of the unit, as well as possible alarms or warning messages. It can also serve as a malfunction diagnostic tool.

The unit was tested in laboratory environment in a small recirculating flume, used for teaching purposes, as depicted in Figure A.3.

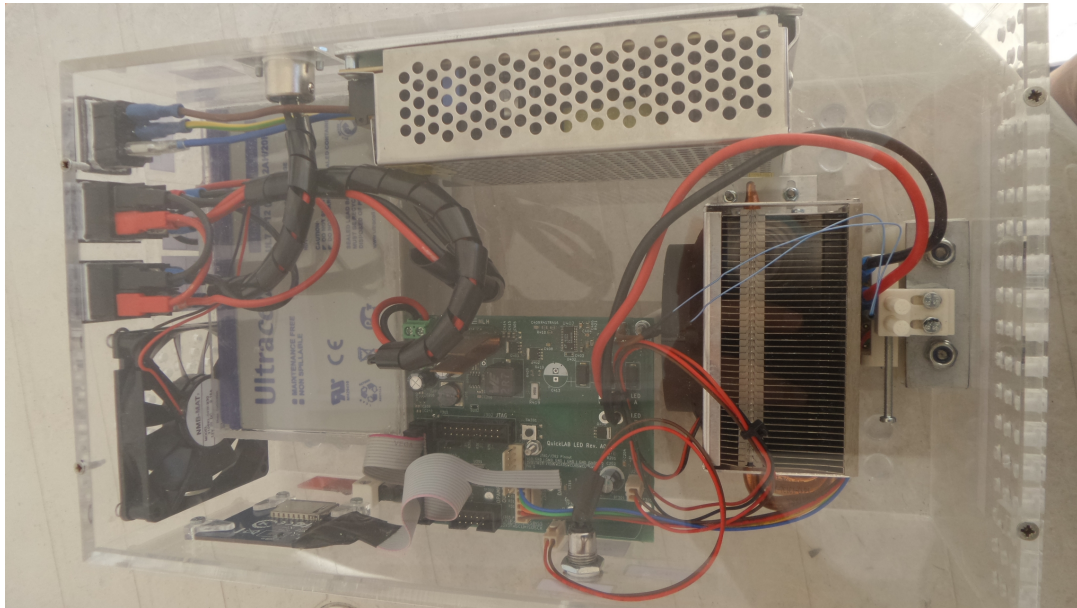


Figure A.1: Portable PIV LED lighting unit.  
Portable PIV LED lighting unit with lead acid battery, control board, LCD, fiber temperature monitor assembly and fiber connection port.

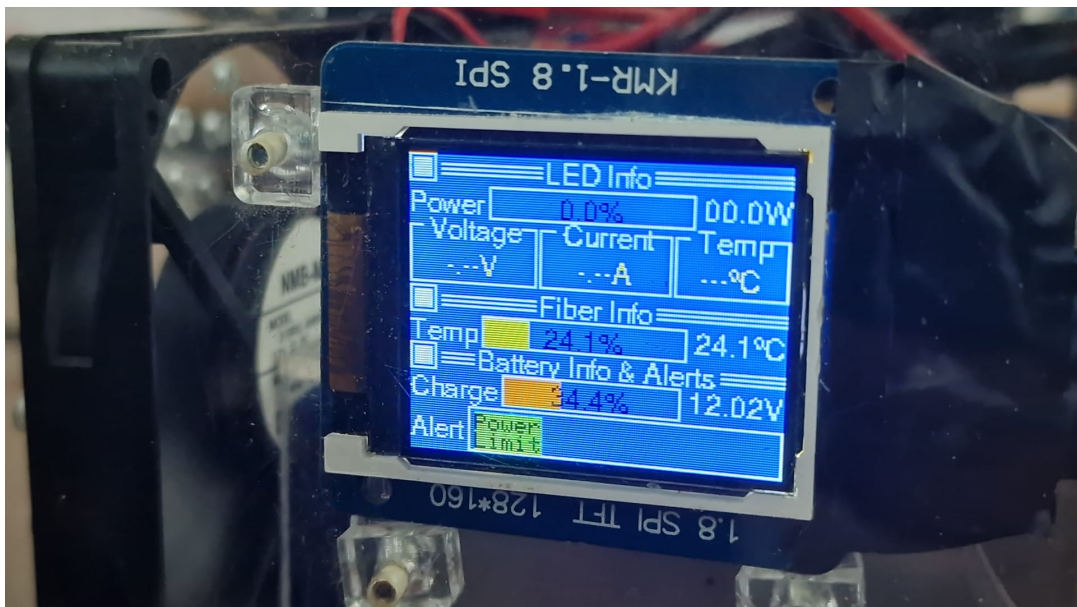


Figure A.2: Portable PIV LED lighting unit - LCD screen.  
Portable PIV LED lighting unit - LCD screen with operation status and information as well as alarms.



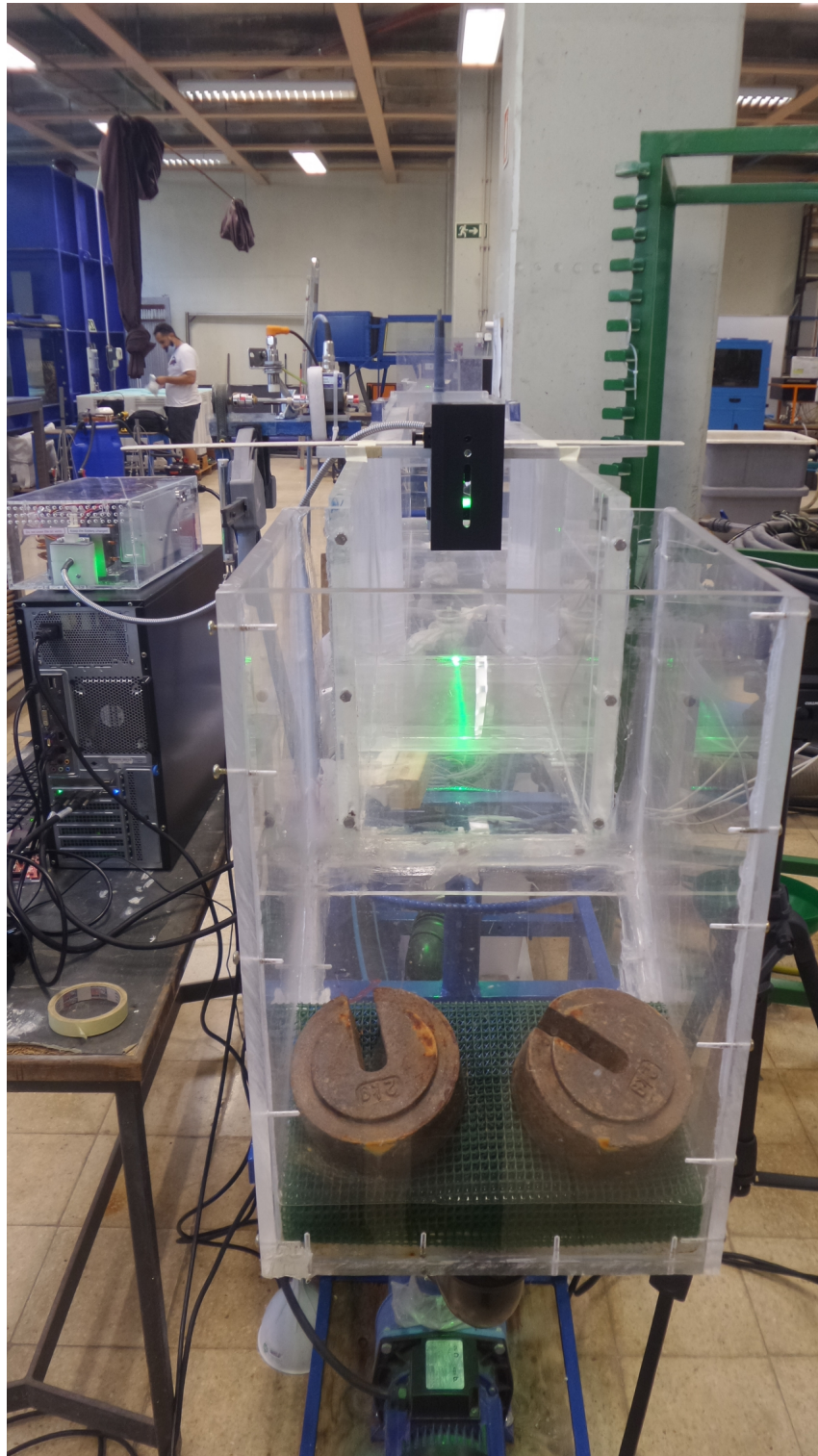


Figure A.3: Portable PIV LED lighting unit in operation.  
Portable PIV LED lighting unit - Unit in operation at the laboratory. The light sheet is visible on the acrylic surface.

## A.2 Experimental PIV data from LED unit

An acquisition with the PIV LED unit was made, composed of 10001 sequential frames at 500 fps, resulting in 10000 PIV image pairs. A sample image obtained in a small recirculating flume with a ramp culminating in a semi closed gate at the exit. A sample PIV image of this PIV image database is shown in Figure A.4.

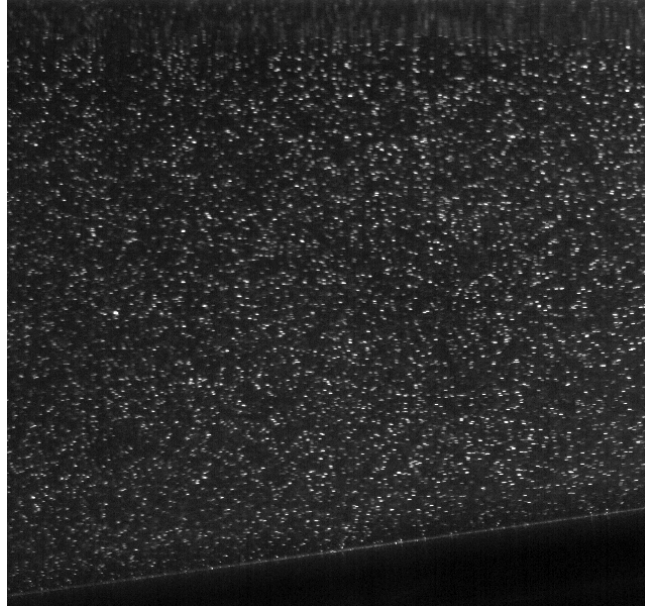


Figure A.4: Sample PIV image from a LED PIV illuminated setup.  
Portable PIV LED lighting unit - PIV image captured by the camera at 500fps and post processed to amplify the brightness level. The image distortion at the top corresponds to the free surface.

The images were then processed by QuickLabPIV-ng in hybrid PIV mode with dense Lucas-Kanade and then post-processed in Python to get the mean velocities of 10000 image pairs and the Reynolds proxies, resulting in the following data, as depicted by the Figures A.5, A.6, A.7, A.8, A.9 and A.10.

## A.3 Conclusions

The portable LED PIV unit was successfully validated for PIV and hybrid PIV applications and is currently in use by a PhD student at UNL for his PIV related work.

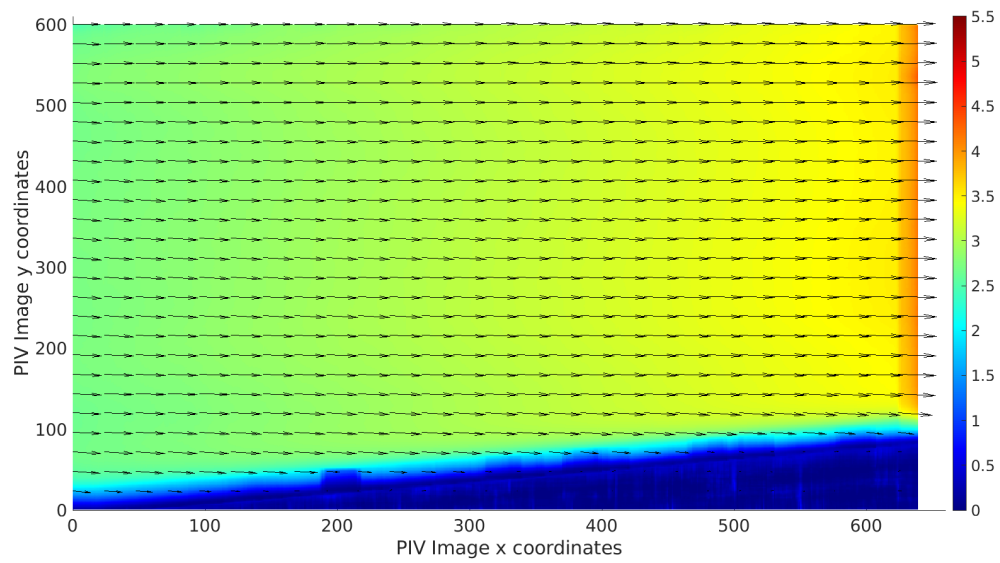


Figure A.5: LED PIV validation - Mean velocity norm with superimposed sparse mean velocity vectors.

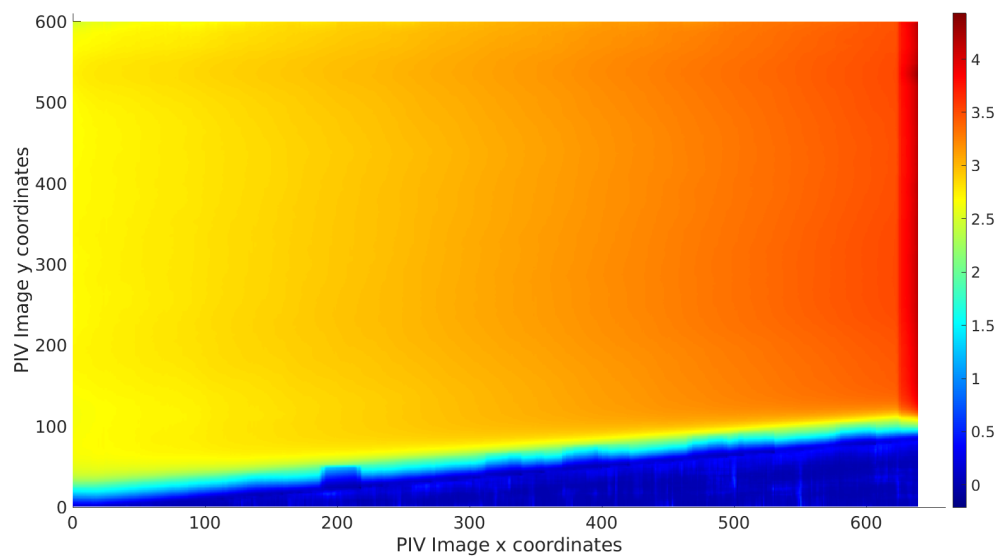


Figure A.6: LED PIV validation - Longitudinal velocity map.

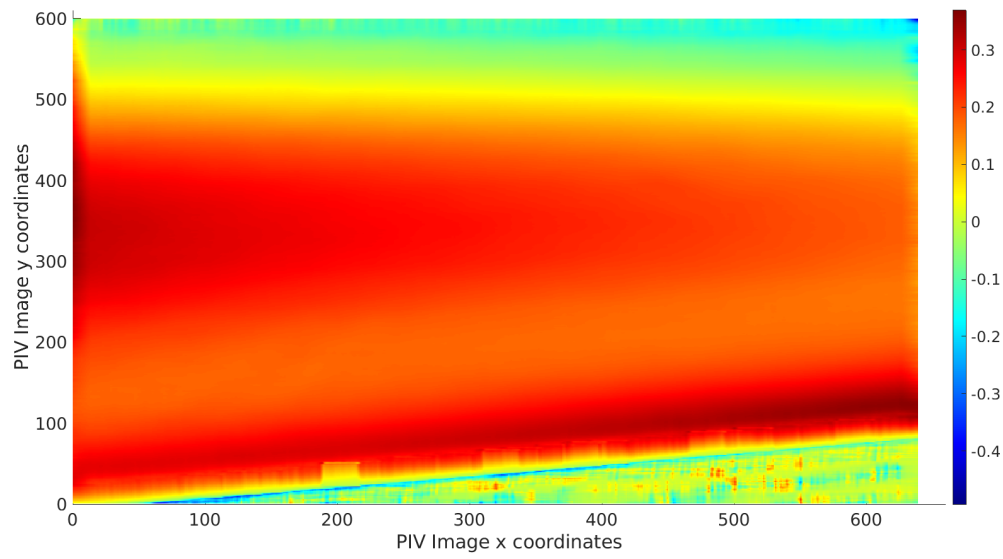


Figure A.7: LED PIV validation - Normal velocity map.

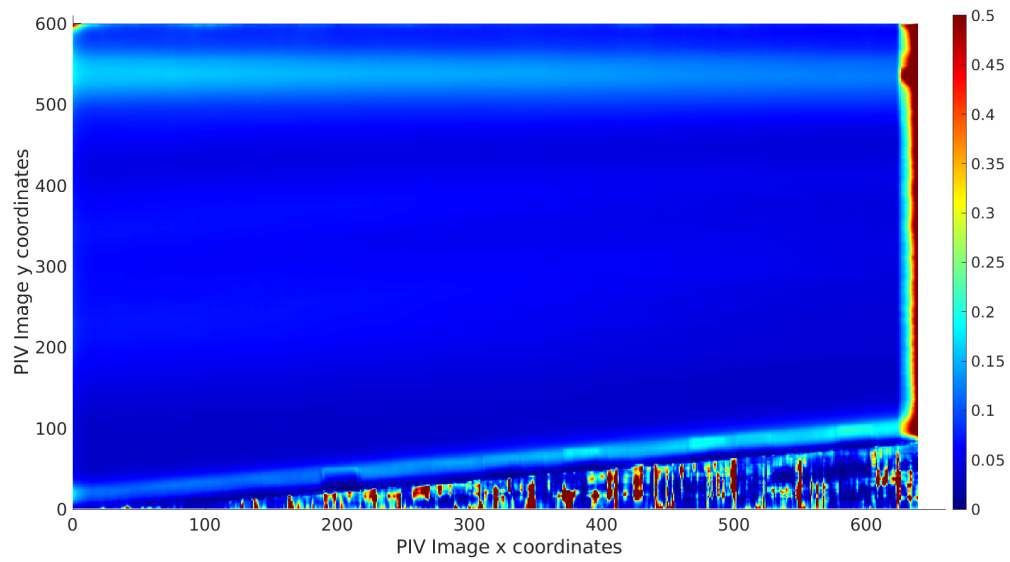


Figure A.8: LED PIV validation - Reynolds proxy for the longitudinal component.

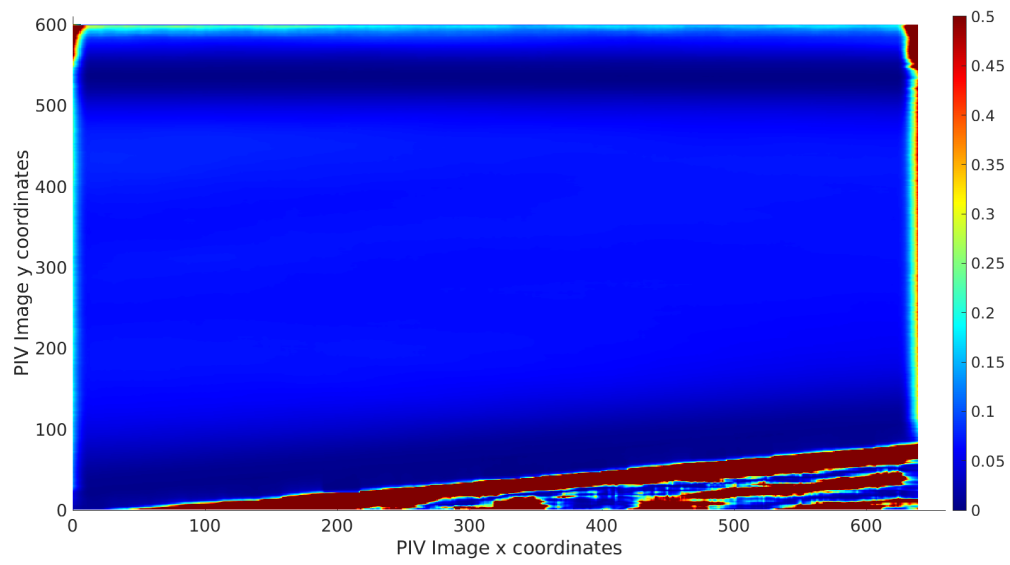


Figure A.9: LED PIV validation - Reynolds proxy for the normal component.

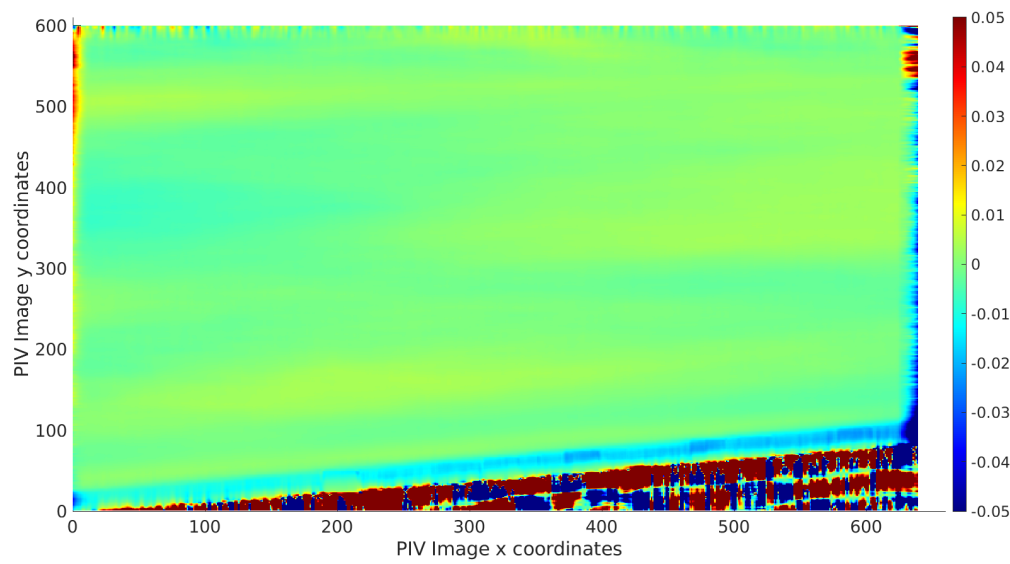


Figure A.10: LED PIV validation - Reynolds proxy for the cross component.



# Bibliography

- [1] L. Mendes, A. Bernardino, and R. M. Ferreira. piv-image-generator: An image generating software package for planar piv and optical flow benchmarking. SoftwareX, 12:100537, **2020**. ISSN 2352-7110. doi: <https://doi.org/10.1016/j.softx.2020.100537>. URL <https://www.sciencedirect.com/science/article/pii/S2352711020300339>.
- [2] L.P.N. Mendes; A.M.C. Ricardo; A.J.M. Bernardino; R.M.L. Ferreira, A comparative study of optical flow methods for fluid mechanics. Exp. Fluids **2021**, 63, 7. doi: <https://doi.org/10.1007/s00348-021-03357-7>.
- [3] L.P.N. Mendes; A.M.C. Ricardo; A.J.M. Bernardino; R.M.L. Ferreira, A Hybrid PIV/Optical Flow Method for Incompressible Turbulent Flows Water **2024**, 16, 1021. doi: <https://doi.org/10.3390/w16071021>.

