

Higher-order accurate neural network for real-time fluid velocimetry

L. Manickathan*, C. Mucignat, I. Lunati

Empa, Swiss Federal Laboratories for Materials Science and Technology, Laboratory of Multiscale Studies in Building Physics, Dübendorf, Switzerland

*Corresponding author: lento.manickathan@empa.ch

Keywords: PIV, CNN, Machine learning

ABSTRACT

In the present work, we introduce a novel lightweight neural network for fluid velocimetry called *LIMA* (Lightweight Image Matching Architecture) designed and optimized for PIV, which can potentially fit on low-cost computer hardware. We compare two versions of the network: *LIMA-4*, a 4-level architecture focused on fast reconstruction; and *LIMA-6*, a 6-level architecture focused on maximizing accuracy. We demonstrate the new approach provides more accurate prediction with fewer network parameters and faster inference speed. Furthermore, we quantified the disparity error using uncertainty quantification (UQ) by image matching to assess the prediction accuracy of the network. We assess the performance of a synthetic direct numerical simulation (DNS) dataset and a wind tunnel measurement dataset of flow past a cylinder. In all cases, we validate that the new network shows higher accuracy than the previous state-of-the-art neural network (*PWCIRR*) and also the classic particle image velocimetry (PIV) approach. In the future, we envision deploying the lightweight architecture on low-cost devices to provide affordable, real-time inference of the flow field during PIV measurements.

Nomenclature

D	Cylinder diameter [m]	$\mathcal{W}$	Warping function
$\mathcal{D}$	Decoder subnetwork	$d\mathbf{s}$	Displacement vector field [px]
$\mathcal{E}$	Encoder subnetwork	$d\mathbf{s}_{\text{ref}}$	Reference displacement vector field [px]
C	Cost volume [-]	l	Network level [-]
$\mathcal{C}$	Cost volume correlation function	t	Time [s]
$\mathbf{F}$	Multi-level feature map [-]	$\mathbf{u}$	Velocity vector field [m s^{-1}]
$\mathbf{I}$	Image intensity [counts]	$\mathbf{x}$	Position vector field [m]
$\mathbf{J}$	Displacement Jacobian matrix [-]	β_1, β_2	Adam optimizer coefficient [-]
L	Number of levels [-]	ϵ	Error [px]
M	Magnification factor [px m^{-1}]	ϵ_d	Disparity error [px]
N	Number of pixels [px]	$\langle \epsilon \rangle$	Spatially-averaged error [px]
$\mathcal{N}$	Neural network	$\langle \epsilon_d \rangle$	Spatially-averaged disparity error [px]
Q	Peak-to-peak ratio [-]	λ_u, λ_J	Trade-off weights [-]
Re	Reynolds number [-]		

1. Introduction

Particle image velocimetry (PIV) is the standard technique used to measure fluid velocity in a non-intrusive manner. Fluid motion is traced using imaging devices that capture the displacement of tracer particles at known time intervals Willert & Gharib (1991). Since the 90s, there has been substantial improvement in hardware and algorithms, and *Window deformation iterative multigrid adaptive resolution* (WIDIM) established itself as the gold standard of PIV post-processing (Scarano, 2002; Schrijer & Scarano, 2008). Recently, with the emergence of novel machine learning (ML) techniques in the field of optical flow for computer vision (Dosovitskiy et al., 2015), there has been a renewed interest in improving fluid velocimetry using non-classical techniques (Cai et al., 2019, 2020; Teed & Deng, 2020).

The keen interest in ML techniques is due to their potential to provide real-time inference speed and recover flow at pixel-level resolution (Hui et al., 2018; Teed & Deng, 2020). In the past, however, ML approaches were not sufficiently advanced or accurate compared to classic PIV techniques. Methods such as fully-connected networks (FCN) (Chen et al., 1998; Rabault et al., 2017) and cascade-style convolutional network (*PIV-DCNN*, Lee et al. (2017)) have shown feasibility for estimating velocity but, due to lack of accuracy and high computational cost, they have not been used extensively. More recently, with the emergence of novel techniques in the field of optical flow for computer vision, there has been a renewed interest in applying ML techniques to velocimetry. Neural networks such as *FlowNet* (Dosovitskiy et al., 2015), *FlowNet2* (Ilg et al., 2017), *LiteFlowNet* (Hui et al., 2018) and *RAFT* (Teed & Deng, 2020) have shown exceptional accuracy and performance with respect to classical techniques. Inspired by these works, CNNs have been applied to fluid velocimetry. Among others we recall works based on *FlowNet* (Cai et al., 2019), *LiteFlowNet* (Cai et al., 2020), *RAFT* (*LightPIVNet* (Yu et al., 2021; Lagemann et al., 2021)), and *PWCIRR* (Manickathan et al., 2022) for PIV.

However, these networks are primarily designed for optical flow estimation and do not achieve, at present, the accuracy of standard PIV post-processing. Moreover, due to the large number of model parameters, they require expensive GPU hardware and are not able to process particle-image pairs in real time. In this paper, we introduce the *Lightweight Image Matching Architecture* (*LIMA*), a novel convolutional neural network (CNN) optimized for PIV, which can potentially fit on low-cost computer hardware, allowing us to achieve higher accuracy than WIDIM and real-time inference.

In this paper, we first give a brief overview of the architecture design, the improvement of the objective function, and the training strategy (section 2). Next, the *LIMA* inference speed is assessed and compared with a state-of-the-art CNN (*PWCIRR*) (section 3.1). In section 3.2, we assess the accuracy of the method with respect to the classic PIV technique and *PWCIRR*. We use both the ground truth and *a-posteriori* uncertainty quantification (UQ) by image matching (Sciacchitano et al., 2013) to quantify the error. Finally, in section 3.3 we present the velocity field reconstructed from an image set from real wind tunnel measurements of flow past a cylinder.

2. Theory

2.1. Lightweight image matching architecture: *LIMA*

A convolutional neural network (CNN) for fluid flow velocimetry estimates the displacement field $\mathbf{ds}$ (px) from the intensity pair $\mathbf{I}$ (counts),

$$\mathbf{ds} = \mathcal{N}(\mathbf{I}), \quad (1)$$

where $\mathbf{I} = (I_1, I_2)^\top$ consists of particle image intensities I_1 and I_2 at time t_1 (s) and $t_2 = t_1 + \Delta t$ (s), respectively. Note that the velocity $\mathbf{u}$ (m s^{-1}) can be readily obtained as $\mathbf{u} = \mathbf{ds}/(\Delta t \cdot M)$, where M is the magnification factor in (px m^{-1}).

As existing CNNs for optical flow estimation entail a large number of parameters and require powerful computational resources, we focused on the design of a lightweight architecture for real-time inference, even on embedded GPU devices, with minimal sacrifice in accuracy. Fig. 1 shows the lightweight image matching architecture *LIMA*. The architecture infers displacement using an iterative residual refinement strategy by Hur & Roth (2019), where the displacement is iteratively corrected and upsampled using l refinement steps. To do so, the encoder subnetwork first constructs an l -level feature map for each image pair,

$$\mathbf{F}_1 = \mathcal{E}(I_1), \quad (2)$$

$$\mathbf{F}_2 = \mathcal{E}(I_2), \quad (3)$$

where $\mathcal{E}$ is the encoder subnetwork, and $\mathbf{F}_1 = (F_1^1, F_1^2, \dots, F_1^L)^\top$ and $\mathbf{F}_2 = (F_2^1, F_2^2, \dots, F_2^L)^\top$ are multi-level pyramid feature maps of the image I_1 and I_2 , respectively. At each refinement step l , the encoded feature from the level l is used to infer the displacement for the level with a specific image resolution. The feature map of image i of level l is warped using displacement estimation from level $l - 1$,

$$\tilde{F}_i^l = \mathcal{W}(F_i^l, \hat{\mathbf{ds}}^{l-1}), \quad (4)$$

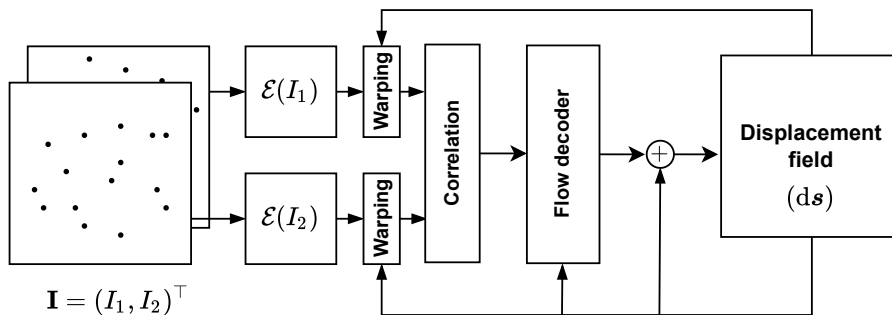


Figure 1. LIMA: Lightweight image matching architecture.

where $\hat{\mathbf{d}}\mathbf{s}$ is $2\times$ upsample of $\mathbf{d}\mathbf{s}$ from level l to match the resolution F_i^l of level l . Note that warping is a non-trainable operator which simply warps the feature map by the displacement map using bilinear interpolation. As the warping is symmetric for I_1 and I_2 , (i.e. towards $t_1 + \Delta t/2$ and $t_2 - \Delta t/2$, respectively), we obtain a central difference image matching that is second-order accurate in time (Wereley & Meinhardt, 2001),

$$\mathbf{u}(t) = \left. \frac{d\mathbf{x}}{d\tau} \right|_{\tau=t} = \frac{\mathbf{x}(t + \Delta t/2) - \mathbf{x}(t - \Delta t/2)}{\Delta t} + \frac{(\Delta t)^2}{24} \left. \frac{d^3\mathbf{x}}{d\tau^3} \right|_{\tau=t} + \dots \quad (5)$$

Then, using both the warped feature map of I_1 and I_2 , a cost volume is constructed to estimate the degree of matching between the two images,

$$C^l = \mathcal{C} \left(\hat{F}_1^l, \hat{F}_2^l \right), \quad (6)$$

where C^l is the cost volume of level l , and $\mathcal{C}$ is the cost volume subnetwork. Finally, with the cost volume of level l and the previous (upsampled) estimate of the displacement from level $l - 1$, we can correct and refine the displacement estimate for level l ,

$$\mathbf{d}\mathbf{s}^l = \hat{\mathbf{d}}\mathbf{s}^{l-1} + \mathcal{D} \left(C^l, \hat{\mathbf{d}}\mathbf{s}^{l-1} \right), \quad (7)$$

where $\mathcal{D}$ is a lightweight decoder subnetwork.

In this study, we train two versions of the *LIMA* architecture: *LIMA-4*, a 4-level version of the network (i.e., $L = 4$) that provides a $1/4^{\text{th}}$ resolution estimate (i.e., 0.25 vector px^{-1}) and; *LIMA-6*, a 6-level network that direct provides a pixel-level inference with the additional 2 refinement steps.

The rationale beyond this is that the 4-level network gives high-throughput inference performance with minor sacrifices in accuracy and a vector field resolution of 4×4 pixels, whereas the 6-level network provides the highest accuracy. Note that *LIMA-4* and *LIMA-6* only differ in the number of level iterations and thus have the same number of trainable parameters due to the weight-sharing design of the iterative residual refinement architecture (Hur & Roth, 2019).

We remark that *LIMA* has considerably fewer of trainable parameters than other CNNs. Indeed, it only has 1.6M parameters, while *PWCIRR* has 3.6M. In contrast, popular networks have 5.3M parameters (*RAFT*, Teed & Deng (2020)), 5.37M parameters (*LiteFlowNet*, Hui et al. (2018)), 38M parameters (*FlowNetS*, Dosovitskiy et al. (2015)), and 162.5M parameters (*FlowNet2*, Ilg et al. (2017)).

2.2. Objective function

The use of novel loss functions is another important improvement. Unlike classic optical flow applications, for which a first-order estimate is sufficient, second-order terms, such as velocity gradients, are also of interest in fluid dynamics. Hence, we define the loss function of the optimization problem such that it incorporates first and second-order quantities of the flow field, i.e.,

$$\mathcal{L} = \sum_{l=1}^L \lambda^l (\lambda_u |\mathbf{d}\mathbf{s} - \mathbf{d}\mathbf{s}_{\text{ref}}| + \lambda_J |\mathbf{J}|), \quad (8)$$

where ds_{ref} is the reference ground truth displacement, ds the estimated displacement, J is the Jacobian of the estimated displacement, λ_u and λ_J are the trade-off weights of displacement and Jacobian, respectively. Notice that the second term of the loss function penalizes fields with large velocity gradients. The loss of each level l is weighted by $\lambda^l = (\lambda^1, \lambda^2, \dots, \lambda^L)$ to control the influence of estimate at a given level.

2.3. Kinematic training strategy

In previous studies, the training dataset consists of solutions of analytical and numerical simulations (Cai et al., 2019, 2020). However, it is computationally expensive to obtain large datasets from such numerical solutions, which require Direct Numerical Simulation (DNS) for a wide set of flow configurations; therefore, only a narrow range of scales and classes of problems are typically used in the training phase. In this study, we opted for the use of examples based on the principle of kinematic training (Manickathan et al., 2022). We demonstrated that such dataset can train the network to efficiently identify the kinematic relationships between two successive tracer particle positions and the corresponding displacement and velocity at a relatively inexpensive computational cost. Notice that this also enables us to arbitrarily increase the dataset size, while the randomization process reduces the risk of over-fitting.

Table 1. Summary of training parameters for *LIMA-4*, *LIMA-6*, and *PWCIRR* (Manickathan et al., 2022).

	<i>LIMA-4</i>	<i>LIMA-6</i>	<i>PWCIRR</i>
Trainable parameters		1.6M	3.6M
Number of levels (L)	4	6	7
Dataset	18278 (kinematic-based) examples		
Total epochs	200		
Batch size	4		
Learning rate	1×10^{-4}		
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)		
Scheduler	ReduceLROnPlateau, decay by 1/5		decay by 1/2 at [60, 80, 100, 120, 140]
Loss function	l_1 -loss, $\lambda_u = 1, \lambda_J = 0.1$		l_2 -loss
Loss weights λ^l	[0.01, 0.02, 0.08, 0.32]	[0.0025, 0.005, 0.01, 0.02, 0.08, 0.32]	[0.32, 0.08, 0.02, 0.01, 0.005, 0.0025, 0.00125]
Training time	24 h (1-node)	104 h (1-node)	123 h (8-node) ($\approx$ 984 h on 1-node)

To improve the generalization of the network, the training dataset is augmented by means of random translation, scaling, rotation, reflection, brightness change, and additive Gaussian noise, which represents image background noise observed during real-world setup. In this study, we use

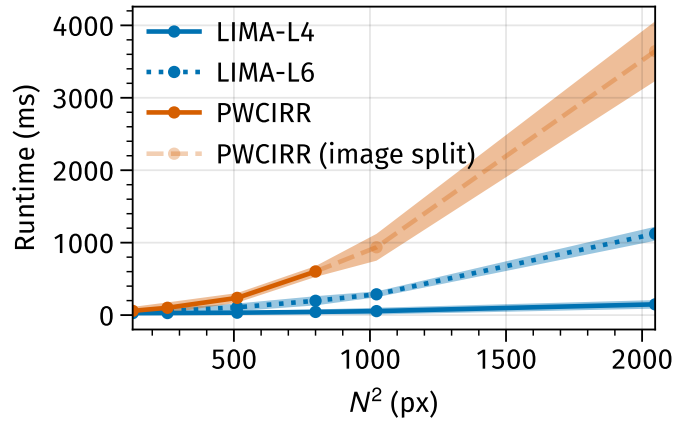


Figure 2. Inference time (ms) for a given square image size N^2 (px): Two versions of *LIMA* (1.6M parameters) (*LIMA-4* and *LIMA-6*) vs. *PWCIRR* (3.6M parameters). Plot shows mean runtime and shaded 3σ . Timings are averaged over 100 runs.

the same data augmentation settings as in previous studies (Cai et al., 2019, 2020; Hui et al., 2018; Ilg et al., 2017). The network is trained according to the parameters tabulated in Tab. 1. The CNN is trained on Tesla P100-PCIE (16 GB) at Swiss National Supercomputing Centre (CSCS).

3. Results and discussions

In this study, we compare the *LIMA-4* and *LIMA-6* architectures with classic *PIV* cross-correlation. We also report the results obtained using another network, *PWCIRR*, trained on the same kinematic dataset.

3.1. Inference time benchmark

First, we benchmark the novel architecture against *PWCIRR* to demonstrate the advantages of the lightweight architecture in terms of computational cost. Fig. 2 shows the inference runtime (ms) for an image size equal to $N \times N$ (px). Note that the runtime performance is benchmarked on Nvidia RTX 2080 Ti and for each image size we provide the average time over 100 runs.

Fig. 2 shows the inference performance of the lightweight network *LIMA* (*LIMA-4*, *LIMA-6*) vs. *PWCIRR*. If we consider an image size of 800×800 px (the maximum image size that *PWCIRR* can analyze directly without splitting), *LIMA-6* and *LIMA-4* are 2.8 times and 19.5 times faster than *PWCIRR*, respectively. Notice that for larger image size, i.e. $N = 2048$, *LIMA-4* and *LIMA-6* have inference runtimes of only 122 ms 1.1 s, respectively, whereas *PWCIRR* requires splitting the input images in smaller packages as the available GPU memory (VRAM) is not enough and thus it has a runtime of 3.64 s. The novel architecture provides exceptional speedup compared to previous state-of-the-art architectures. In particular, *LIMA-4* is so fast that could be used for real-time inference on embedded GPU devices.

3.2. Accuracy assessment on a synthetic DNS dataset

To assess the accuracy of the novel architecture, we compare the estimation provided by *LIMA-4* and *LIMA-6* with *PWCIRR* as well as with standard *PIV* on a synthetic image pair in which the tracers move according to a velocity field obtained by *Direct Numerical Simulation* (DNS) (Carlier, 2005). Notice that we trained the networks on a kinematic random dataset and this type of data has not been shown to the networks. Fig. 3 reports the distribution of the displacement magnitude $|ds|$ (px) of DNS synthetic images comparing the different cases. Although *PIV* data and *LIMA-4* provide an area-averaged estimation whereas the other methods have pixel-level resolution, all calculated displacements show similar patterns.

The vertical profile at $x = 128$ px, Fig. 3(g). shows that *PWCIRR* and *LIMA-6* match more faithfully the reference displacement profiles, while *LIMA-4* and *PIV* occasionally deviate away from the reference especially in regions near high velocity gradients (i.e., $y = 110$, $y = 200$, and $y = 225$).

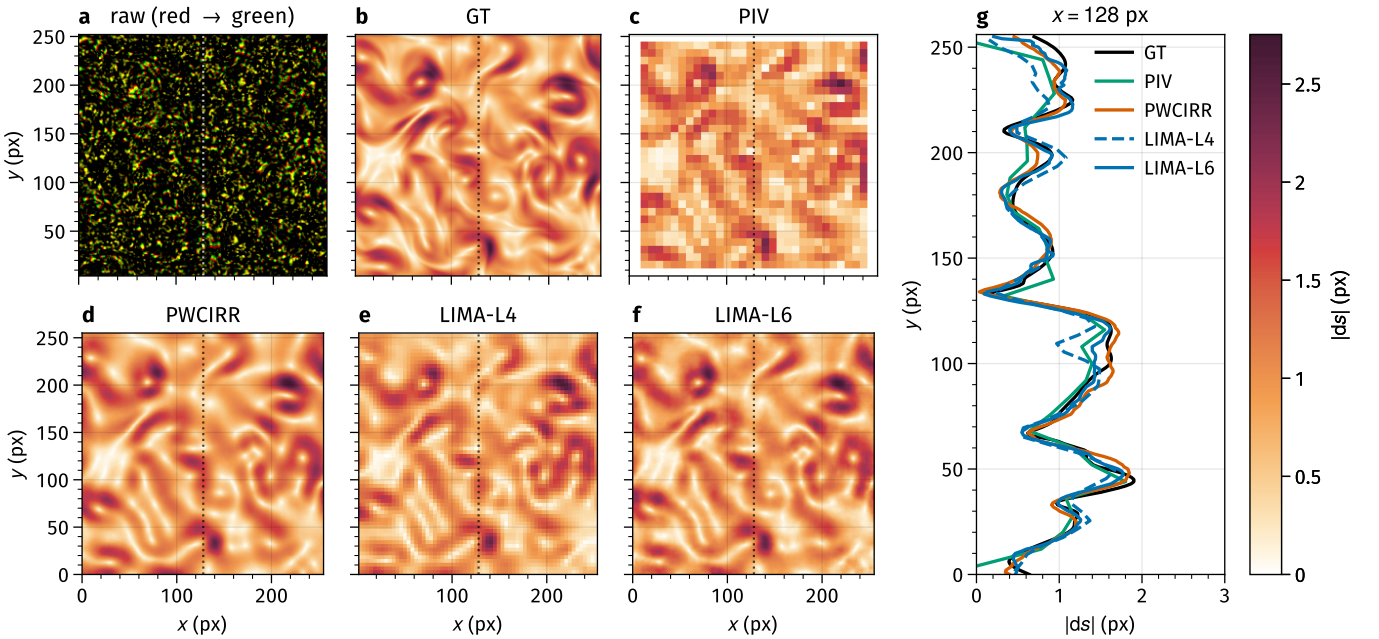


Figure 3. Displacement magnitude $|ds|$ (px) of synthetic PIV images obtained from Carlier (2005): (a) raw image (red: I_1 ; green: I_2); (b) ground-truth (GT); (c) *PIV* (8 × 8 px); (d) *PWCIRR*; (e) *LIMA-4* (L = 4); (f) *LIMA-6* (L = 6); (g) vertical profile along $x = 128$ px.

To quantify the estimation error, we use two approaches. First, we directly calculate the error ϵ (px) from the reference ground truth,

$$\epsilon = |ds - ds_{\text{ref}}|, \quad (9)$$

where ds_{ref} is the reference map and ds is the estimate; then we use the *a-posteriori* disparity error ϵ_d (px) by image matching (Sciacchitano et al., 2013) to estimate the uncertainty of the displacement estimate. Fig. 4 compares both errors of *PIV* and *PWCIRR* with *LIMA-4* and *LIMA-6*. We also quantify the spatially-averaged error $\langle \epsilon \rangle$ and disparity error $\langle \epsilon_d \rangle$. To minimize boundary artifacts

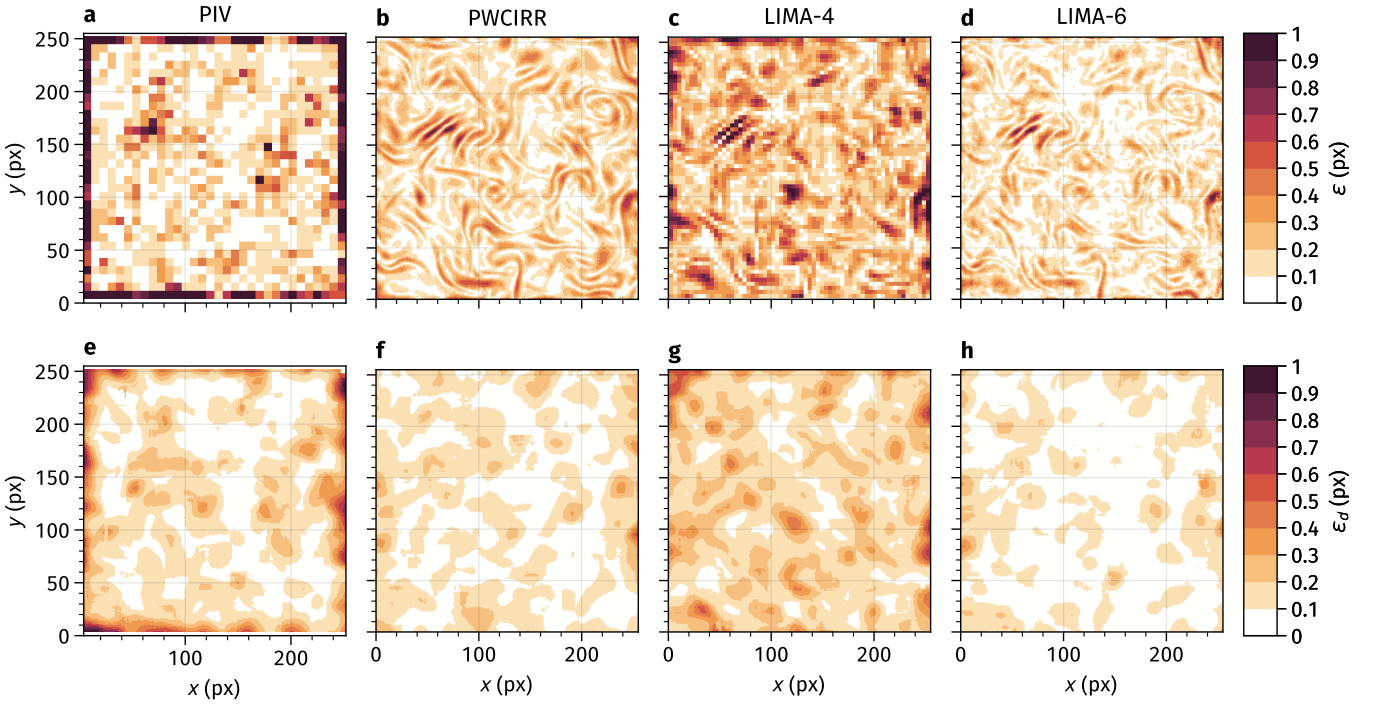


Figure 4. Uncertainty quantification of DNS images: (a-d) Displacement error magnitude with respect to the ground truth, $\epsilon = |\mathbf{ds}_{\text{ref}} - \mathbf{ds}|$ (px); (e-h) Pixelwise disparity error ϵ_d (px) using *a-posteriori* UQ by image matching (Sciaccchitano et al., 2013). Compares: (a, e) *PIV* (8×8); (b, f) *PWCIRR*; (c, g) *LIMA-4* ($l = 4$); (d, h) *LIMA-6* ($l = 6$).

the spatial average is performed excluding velocity vectors closer than 8 px from the boundaries. Comparing the results in Tab. 2, we observe that *LIMA-6* provides the highest accuracy with $\langle \epsilon \rangle = 0.172$ px and $\langle \epsilon_d \rangle = 0.104$ px. On the contrary, *LIMA-4* has the least accuracy with $\langle \epsilon \rangle = 0.306$ px and $\langle \epsilon_d \rangle = 0.203$ px. This suggests that the number of levels (or iterations) is key to provide accurate estimate, but more levels require additional computations.

Table 2. Spatially-averaged displacement error $\langle \epsilon \rangle$ (px) and pixelwise disparity error $\langle \epsilon_d \rangle$ (px). The spatial average is performed excluding velocity vectors closer than 8 px from the boundaries.

Method	$\langle \epsilon \rangle$ (px)	$\langle \epsilon_d \rangle$ (px)
<i>PIV</i>	0.205	0.194
<i>PWCIRR</i>	0.205	0.123
<i>LIMA-4</i>	0.306	0.203
<i>LIMA-6</i>	0.172	0.104

3.3. Experimental study of flow past a cylinder

Finally, we assess the performance of *LIMA* with real data from a wind tunnel experiment of flow past a cylinder (Manickathan et al., 2022). The experiment is performed in the wind tunnel of the *Swiss Federal Laboratories for Materials Science and Technology* (Empa). The facility has a cross-section

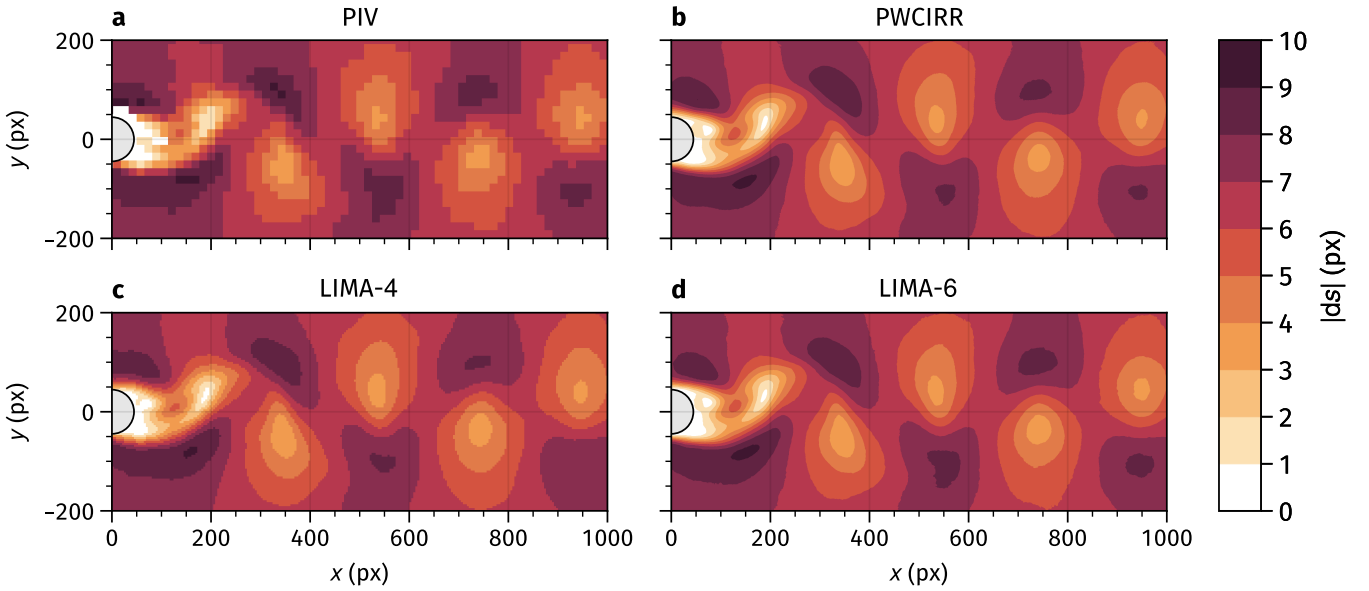


Figure 5. Displacement magnitude $|ds|$ (px) of cylinder wake: (a) *PIV* (32×32 , 50% ov.); (b) *PWCIRR*; (c) *LIMA-4* ($L = 4$); *LIMA-6* ($L = 6$).

of $1.9 \text{ m} \times 1.3 \text{ m}$ (width and height, respectively) and can operate with a bulk wind speed u_∞ ranging from 0.5 to 25 m s^{-1} . The imaging setup consists of Nd:YLF dual cavity laser with a pulse energy of 30 mJ at 1 kHz and 4 Mpx high-speed camera of $2016 \text{ px} \times 2016 \text{ px}$. By using a 200 mm focal lens (numerical aperture equal to $f/2.8$) in combination with a $2\times$ teleconverter a magnification factor M equal to 15 px mm^{-1} was achieved. The flow is seeded using Di-Ethyl-Hexyl-Sebacat (DEHS) tracer particles of nominal size $1 \text{ }\mu\text{m}$ obtained with a lasking seeding generator.

The flow past the cylinder of diameter $D = 6 \text{ mm}$ is measured at 0.47 m s^{-1} , which correspond to a Reynolds number $\text{Re} = 189$. The separation time is chosen to obtain a maximum displacement of 7 px in the free stream. A standard PIV analysis is performed with *WIDIM* (Scarano, 2002; Schrijer & Scarano, 2008) including background subtraction, an initial interrogation grid of $64 \text{ px} \times 64 \text{ px}$, and a refinement step of $32 \text{ px} \times 32 \text{ px}$ with 50% overlap. Vector validation was performed based on peak-to-peak ratio $Q < 1.25$ and a NMF with 3×3 kernel.

Fig. 5 shows the magnitude of the displacement $|ds|$ (px) in the cylinder wake obtained with *PIV*, *PWCIRR*, *LIMA-4*, and *LIMA-6* at $\text{Re} = 189$. All approaches provide similar estimates of the flow in the cylinder wake. Minor differences are observed in the near-wake region, i.e., $x = [50, 350] \text{ px}$ and $y = [-100, 100] \text{ px}$. The PIV vector field contains artifacts in the vicinity of the cylinder, possibly due to the low signal-to-noise ratio induced by laser reflections. In contrast, both *LIMA* and *PWCIRR* are able to reconstruct the flow field in this region, depicting distinct flow features and realistic boundary layer separation from the cylinder.

For a more rigorous assessment of the errors, we employ the uncertainty quantification (UQ) by image matching (Sciacchitano et al., 2013). Fig. 6 shows the disparity error ϵ_d (px) in the cylinder wake for *PIV*, *PWCIRR*, *LIMA-4* and *LIMA-6*. All methods perform very similarly, with minor

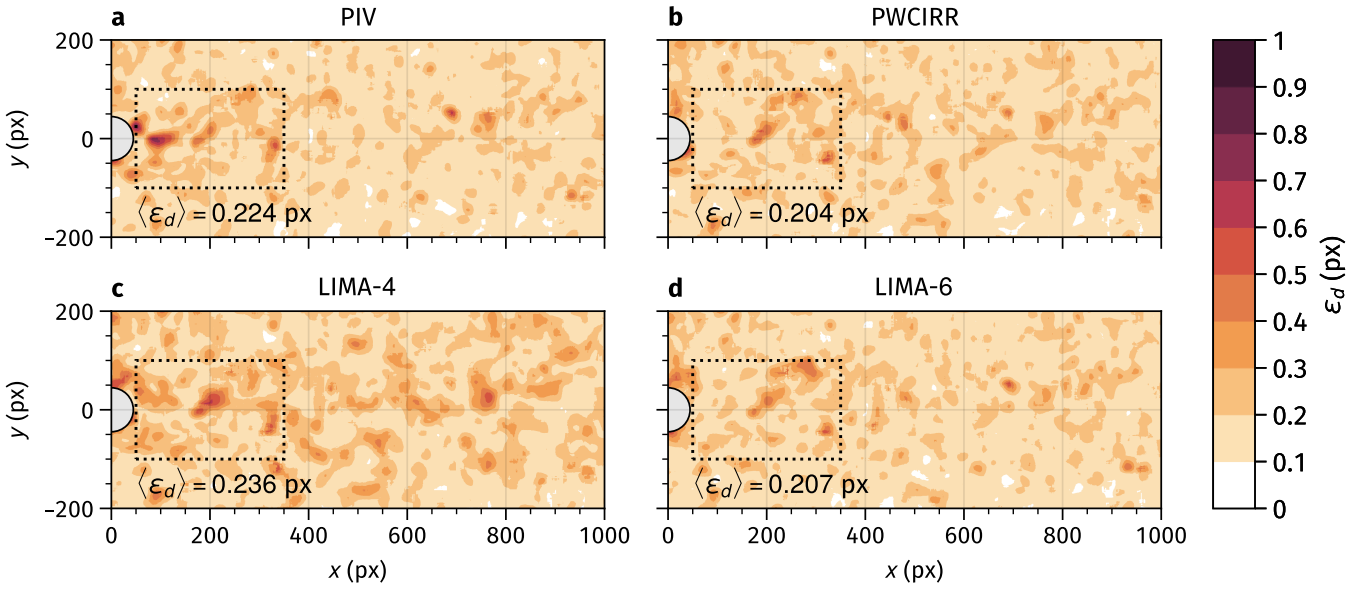


Figure 6. Pixelwise disparity error map ϵ_d of cylinder wake using *a-posteriori* UQ by image matching (Sciacchitano et al., 2013): (a) *PIV*; (b) *PWCIRR*; (c) *LIMA-4* ($L = 4$); *LIMA-6* ($L = 6$). The near-wake region is indicated by the bounding box $x = [50, 350]$ px and $y = [-100, 100]$ px along with the spatially-averaged disparity error $\langle \epsilon_d \rangle$.

discrepancies only in the near-wake region. *PIV* estimate shows the highest error in the near-wake, similar to *LIMA-4*, which also has higher errors in the wake compared to other ML estimates. *PWCIRR* and *LIMA-6* show the best accuracy: the area-averaged disparity error in the near-wake region (indicated by the dotted boundary box in Fig. 6) is $\langle \epsilon_d \rangle = 0.204$ px and $\langle \epsilon_d \rangle = 0.207$ for *PWCIRR* and *LIMA-4*, respectively, whereas we have $\langle \epsilon_d \rangle = 0.224$ px for *PIV* and $\langle \epsilon_d \rangle = 0.236$ px for *LIMA-4*.

4. Conclusions

We devised a novel architecture specifically designed for real-time inference: the Lightweight image matching architecture *LIMA*. Unlike standard optical flow-based networks, we employ symmetric image matching to ultimately obtain a second-order accurate estimate of the displacement field. Furthermore, we incorporate a second-order gradient term into the loss function in order to improve the accuracy of the flow reconstruction. We compare two versions of the network: *LIMA-4*, a 4-level architecture focused on fast reconstruction; and *LIMA-6*, a 6-level architecture focused on maximizing accuracy. *LIMA-4* is 19.5 times faster than the state-of-the-art network (*PWCIRR*), and even *LIMA-6* provides a 2.25 times speed while achieving higher accuracy. A rigorous uncertainty quantification (UQ) was performed by employing disparity error assessment using image matching for a synthetic dataset obtained by direct numerical simulation (DNS) and for data from a wind tunnel experiment. The novel network not only shows promising speed up but also higher accuracy than the state-of-the-art *PWCIRR*.

We remark that the two versions are complementary and can be jointly employed. For instance,

LIMA-4 could be used during the experiments to inspect the flow in real time and obtain a quick assessment, while *LIMA-6* could be used later on, off-line, to reconstruct the flow with the highest accuracy.

We envision to deploy the lightweight architecture on low-cost devices to provide an affordable real-time inference of the flow field *in-situ* during PIV measurements. Possible future extensions include the use of mobile phone cameras and reduced data acquisition when the measurement matrix becomes intractable in terms of computational time or storage need.

Acknowledgements

We would like to thank Stephan Kunz, Roger Vonbank, and Beat Margelisch for their support at Empa for the experimental setup. We acknowledge access to *Piz Daint* at the Swiss National Supercomputing Centre, Switzerland, under Empa's share with the project ID: *em13*. Finally, we thank Dr. Hossein Gorji for fruitful discussions.

References

- Cai, S., Liang, J., Gao, Q., Xu, C., & Wei, R. (2020, 6). Particle Image Velocimetry Based on a Deep Learning Motion Estimator. *IEEE Transactions on Instrumentation and Measurement*, 69(6), 3538–3554. Retrieved from <https://ieeexplore.ieee.org/document/8793167/>
- Cai, S., Zhou, S., Xu, C., & Gao, Q. (2019, 4). Dense motion estimation of particle images via a convolutional neural network. *Experiments in Fluids*, 60(4), 73. Retrieved from <http://dx.doi.org/10.1007/s00348-019-2717-2>
- Carlier, J. (2005). Second set of fluid mechanics image sequences. *European Project Fluid Image Analysis and Description (FLUID)*-<http://www.fluid.irisa.fr>, 0018–9456.
- Chen, P.-H., Yen, J.-Y., & Chen, J.-L. (1998, 5). An artificial neural network for double exposure PIV image analysis. *Experiments in Fluids*, 24(5-6), 373–374. doi:
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., ... Brox, T. (2015, 12). FlowNet: Learning Optical Flow with Convolutional Networks. In *2015 IEEE international conference on computer vision (iccv)* (Vol. 2015 Inter, pp. 2758–2766). IEEE. Retrieved from <https://ieeexplore.ieee.org/document/7410673/> doi:
- Hui, T.-W., Tang, X., & Loy, C. C. (2018, 5). LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8981–8989.

- Hur, J., & Roth, S. (2019). Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (p. 5747-5756). doi:
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (p. 1647-1655). doi:
- Lagemann, C., Lagemann, K., Mukherjee, S., & Schröder, W. (2021). Deep recurrent optical flow learning for particle image velocimetry data. *Nature Machine Intelligence*, 3(7), 641–651. Retrieved from <http://dx.doi.org/10.1038/s42256-021-00369-0> doi:
- Lee, Y., Yang, H., & Yin, Z. (2017, 12). PIV-DCNN: cascaded deep convolutional neural networks for particle image velocimetry. *Experiments in Fluids*, 58(12), 171. Retrieved from <http://dx.doi.org/10.1007/s00348-017-2456-1><http://link.springer.com/10.1007/s00348-017-2456-1> doi:
- Manickathan, L., Mucignat, C., & Lunati, I. (2022). Kinematic training of Neural Network for Particle Image Velocimetry. *Measurement Science and Technology* (under review).
- Rabault, J., Kolaas, J., & Jensen, A. (2017, 12). Performing particle image velocimetry using artificial neural networks: a proof-of-concept. *Measurement Science and Technology*, 28(12), 125301. Retrieved from <http://stacks.iop.org/0957-0233/28/i=12/a=125301?key=crossref.3d6dce24a30c239c2dbbcfd39681a6c9> doi:
- Scarano, F. (2002, jan). Iterative image deformation methods in PIV. *Measurement Science and Technology*, 13(1), R1–R19. Retrieved from <https://iopscience.iop.org/article/10.1088/0957-0233/13/1/201> doi:
- Schrijer, F. F. J., & Scarano, F. (2008, 11). Effect of predictor–corrector filtering on the stability and spatial resolution of iterative PIV interrogation. *Experiments in Fluids*, 45(5), 927–941. Retrieved from <http://link.springer.com/10.1007/s00348-008-0511-7> doi:
- Sciacchitano, A., Wieneke, B., & Scarano, F. (2013). PIV Uncertainty Quantification by Image Matching. *Measurement Science and Technology*, 24(4). doi:
- Teed, Z., & Deng, J. (2020). *Raft: Recurrent all-pairs field transforms for optical flow*. arXiv. Retrieved from <https://arxiv.org/abs/2003.12039> doi:
- Wereley, S. T., & Meinhart, C. D. (2001, 9). Second-order accurate particle image velocimetry. *Experiments in Fluids*, 31(3), 258–268. Retrieved from <http://link.springer.com/10.1007/s003480100281> doi:
- Willert, C. E., & Gharib, M. (1991). Digital particle image velocimetry. *Experiments in Fluids*, 10(4), 181–193. Retrieved from <https://doi.org/10.1007/BF00190388> doi:

Yu, C., Bi, X., Fan, Y., Han, Y., & Kuai, Y. (2021). LightPIVNet: An Effective Convolutional Neural Network for Particle Image Velocimetry. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–15. Retrieved from <https://ieeexplore.ieee.org/document/9437223/> doi: