

TUBflow - An Open Source Application for Digital Postprocessing of Oil Film Visualizations in Wind Tunnels

Lennart Rohlfs ^{*}, Léo Max Hastrich [†], Alex Gothow [‡], and Julien Weiss [§]

Technical University of Berlin, 10587 Berlin, Germany

We present TUBflow, an innovative open-source application for the digital postprocessing of oil film visualizations in wind tunnel tests, addressing the limitations of traditional analog techniques. The proposed method leverages modern computer vision algorithms, specifically optical flow techniques, to automate the extraction of skin friction lines from a sequence of oil-film images. The user-friendly graphical user interface (GUI) of TUBflow allows easy pre-processing of the input images and visualization of the results, while the OpenCV library enables the efficient implementation of several optical flow algorithms. The versatility of the approach is demonstrated by applying it to three different wind-tunnel test cases: 1) A subsonic half-diffuser flow, 2) An incident shock-wave/boundary-layer interaction, and 3) The suction side of a high-lift wing with distributed propulsion. The results show that TUBflow enhances the interpretation of oil-film visualizations by effectively capturing complex flow structures across the image sequences. This makes it a promising tool to aid the digital post-processing of oil-film visualizations in ground-testing campaigns. The program and the code accompanying this manuscript are made publicly available at <https://github.com/AeroTUBerlin/TUBflow>.

I. Introduction

Surface flow visualization is a classical technique in wind tunnel testing. It is used by test engineers to discover relevant flow patterns on aerodynamic surfaces and to help interpret quantitative force-and-moment data. While the use of tufts is probably the simplest practice, the oil-film technique is often preferred because of its high sensitivity to local changes in the boundary layer, enabling for example the detection of laminar-turbulent transition or the determination of two- or three-dimensional separation and reattachment lines [1].

Oil-film visualization is typically performed by coating the model with a specifically prepared paint consisting of a suitable oil and a finely powdered pigment. When the air flow is turned on, the stream carries the oil with it, and the remaining streaky deposit of the pigment gives an information on the direction of the flow [2]. In a stationary flow, taking a picture of the resulting streaks enables the experimentalist to draw skin-friction lines and thereby estimate the near-wall flow topology. In recent examples, several authors experimentally characterized the topology of turbulent separated flows in finite-span wind tunnels [3, 4]. Examples of oil-film visualization on aircraft wings are provided in the book by Barlow [1]. When the oil is kerosene and the powder is kaolin, the technique has traditionally been called “China clay” in the ground-testing community [1].

A drawback of the classical oil-film visualization is its analog character. Skin-friction lines are typically sketched by hand on an image of the oil film to interpret the flow topology (e.g. [3, 4]). This limits the efficiency of the data analysis and hinders the reporting and dissemination of the results. Furthermore, the analysis remains qualitative, as only few skin friction lines are typically drawn, although the raw images inherently contain information about the entire shear stress field on the surface.

To extract further information and extend the capability of surface flow visualizations, the motion of the oil film on the surface can be modeled and linked to the brightness levels of the recorded image sequence as demonstrated by Liu and Shen [5]. Interestingly, the derived equations take the form of an optical flow problem and thus can be solved with dedicated algorithms. For example, the Global Luminescent Oil-Film (GLOF) technique of Liu et al [6] implements a modified variant of the Horn-Schunck algorithm [7] for optical flow problems and demonstrates its capability to extract normalized shear stress fields from a series of luminescent oil-film images. Similar methods that use other measurable surface quantities are described in the review article by Liu [8]. A potential drawback of the GLOF technique is that it

^{*}Graduate Research Assistant, Chair of Aerodynamics, Institute of Aerodynamics and Astronautics

[†]Master Student, Chair of Aerodynamics, Institute of Aerodynamics and Astronautics

[‡]Graduate Research Assistant, Chair of Aerodynamics, Institute of Aerodynamics and Astronautics, Student Member AIAA

[§]Professor, Chair of Aerodynamics, Institute of Aerodynamics and Astronautics, Associate Fellow AIAA

requires prior knowledge of additional parameters such as the oil-film thickness and luminescence which is difficult to estimate in a typical wind tunnel experiment.

In the current paper, we propose an intermediate method to digitally extract skin friction lines from a sequence of oil-film images taken in the wind tunnel. In contrast to GLOF, we do not seek to obtain an estimate of the shear-stress field, but merely a digital plot of the wall streamlines using images as input. Hence, our goal is to automate the manual drawing of skin friction lines from usual oil-film images. The proposed method is based on a direct solution of the optical flow equation using a sequence of oil-film images as input. We take advantage of modern algorithms implemented in the Open CV computer vision library in Python to produce a simple, but versatile tool that may be used to enhance existing oil-film data, or aid in the interpretation of future results in ground-testing campaigns. The paper is structured as follows: First, we provide some theoretical background on the optical flow equation and the algorithms used to solve it. Second, we introduce our implementation into a user friendly graphical user interface (GUI). Finally, we apply our method to sequences of oil-film images obtained in three different wind-tunnel tests to demonstrate the versatility of the approach and provide some best-practices.

II. Theoretical Background

During an oil-film visualization, the oil constantly moves over the model surface following the local shear stress vector. Over time, more reflective oil will accumulate in areas with low shear stress, resulting in a higher luminescent intensity in the recordings and vice versa. Thus, by analyzing the change in luminescent intensity over a series of images as an optical flow problem, the qualitative shear stress field can be reconstructed.

The optical flow technique is widely used in the field of computer vision and is based on the idea that for constant illumination the brightness emitted by a moving object remains constant and thus can be related to its velocity by a transport equation:

$$\frac{\partial I(\vec{x}, t)}{\partial t} + \vec{v}(\vec{x}, t) \cdot \vec{\nabla} I(\vec{x}, t) = 0 \quad (1)$$

Here I is the recorded intensity or brightness of the image and \vec{v} is the velocity of each image point in space and time. This equation has the same form as that derived by Liu and Shen [6] for fluid motion in an oil film, except for an additional small term on the right-hand side that may often be neglected. Assuming that the time between consecutive frames is constant, as in a video or timelapse recording, Eq.1 can be simplified to:

$$I_0(\vec{x}) - I_1(\vec{x} + \vec{v}(\vec{x}, t)) = 0. \quad (2)$$

This equation is also known as the displaced frame difference equation (DFD). For small perturbations, the optical flow equation (1) can be linearized and further simplified to

$$I_x u + I_y v + I_t = 0 \quad (3)$$

where I_x and I_y are the spatial gradients and I_t is the temporal intensity difference between frames. Solving Eq. 3 for (u, v) is an underdetermined problem that requires some form of closure. Since its introduction by Horn and Schunck [7] in the 1980s, many different algorithms have been proposed to solve the optical flow problem. A common approach is to minimize an energy functional combining a data term J_D containing Eq. 3 with a weighted regularization term J_R that enforces smoothness on the velocity field since neighbouring pixels will typically move in a similar direction:

$$\vec{\delta} = \underset{\vec{\delta}}{\operatorname{argmin}} J_D(I_0, I_1, \vec{\delta}) + \lambda J_R(\vec{\delta}) \quad (4)$$

In their original paper, Horn and Schunck use the sum of the first order gradients of the velocity field for regularization, which yields very smooth and consistent motion fields. While this approach is valid for tracking moving objects or people, it struggles with discontinuities that arise for example at boundaries and dampens the magnitude of the entire gradient field [9]. These issues can be overcome by using regularization terms that are based on higher order gradients as demonstrated by [10, 11].

Farnebäck [12] proposed another popular algorithm that solves the optical flow problem by approximating the motion between two consecutive frames using polynomial expansion. For each image pair, the intensity field of the first frame is approximated by a quadratic polynomial

$$I_0(x, y) \approx a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 x y + a_5 y^2 \quad (5)$$

where a_n are the polynomial coefficients. The shifted intensity field $I_1(x + u, y + v)$ is then approximated by the same polynomial with shifted coordinates. To handle larger displacements and improve robustness, the Farnebäck algorithm additionally uses a multi-scale approach that constructs an image pyramid where each level is a downsampled version of the previous one. At each level of the pyramid, the algorithm iteratively refines the flow field by minimizing the difference between the polynomial approximations of the two frames. This makes the Farnebäck method computationally very efficient but more sensitive to noise compared to classical variational methods.

Another method that can predict the optical flow very efficiently is the Dense Inverse Search (DIS) algorithm that was proposed recently by Kroeger et al [13]. Similar to the Farnebäck algorithm, it also uses a multi-scale approach while trying to find correspondences between patches in consecutive frames using an inverse compositional image alignment approach. While the idea of matching patches or shapes instead of single points is used heavily in sparse optical flow algorithms such as the Lucas-Kanade method [14], the required computations are typically too expensive for dense calculations. In contrast, the DIS algorithm uses a much more efficient inverse approach that does not require gradient computations at each iteration step and is thus much faster.

III. Implementation

In this section we introduce the different modules that TUBflow is composed of. The motivation behind the program structure is to make it user-friendly with few and essential steps, but it shall also allow flexibility. It makes heavy use of the OpenCV package in python that contains all the required tools for the image pre-processing as well as numerous optical flow solvers including the previously introduced Farnebäck and DIS algorithms. It is therefore possible to create an automatic and efficient process to extract the flow out of oil-film visualization recordings. Figure 1 shows a flowchart of the program.

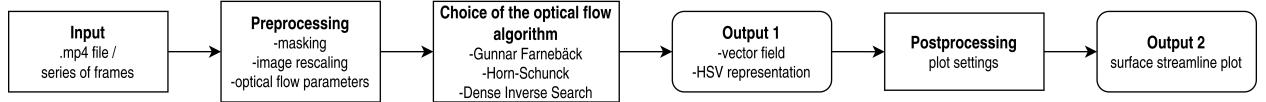


Fig. 1 Structure of the TUBflow program

The program is an executable file which launches a GUI. The user first gives a movie file or a series of frames as input. Since the optical flow is calculated for each pixel, the required computational power significantly increases with higher image resolutions. For this reason, an option was added to resize the images via area-interpolation in advance with a scaling factor λ_{scale} . Also, as will be discussed later, decreasing the image resolution often actually improves quality of the computed streamlines, as image noise is reduced. Additionally, multiple masks can be generated and saved, so that only relevant pixels are used in the calculation. For this task, we created an interactive interface where the user selects at least 3 points on the image. A mask is then generated by connecting the points to a polygon and either the points inside or outside the mask are blacked out.

After the preprocessing, it is then possible to choose between the previously introduced optical flow algorithms: Gunnar-Farnebäck, Horn-Schunck and Dense Inverse Search. To allow only specific parts of the recording to be taken into account for the optical flow calculation, the start and end frame can be changed. The interval between each frame can also be increased, resulting in shorter runtimes whilst decreasing the precision of the result. Furthermore, the implementation of the Gunnar-Farnebäck method in OpenCV allows to change optical flow parameters such as the size of the image pyramids or the number of iterations. Adapting them to the test configuration can lead to better results, but can impact the computing time. The calculation can then be executed and returns a two-dimensional array with size of the images over which the optical flow is calculated. While in rare cases the information of only two images suffices to recreate the entire flow field, much better results are obtained when calculating the optical flow over a longer sequence of images. This allows the algorithm to average the results and thus produce a smoother representation of the flow field. Let F denote the optical flow of a single image pair, which is calculated by a function G with the variables img_1 and img_2 .

$$F = G(img_1, img_2) \quad (6)$$

The average optical flow F_{avg} over n images is calculated from the sum of all optical flows between consecutive pairs of images as follows:

$$F_{avg} = \frac{1}{n-1} \sum_{i=2}^n G(img_{i-1}, img_i) \quad (7)$$

Equation 7 holds only if the optical flow is constant in the time interval, which is a reasonable assumption if the flow is statistically stationary, and if the moving oil film is observed over a sufficiently long time interval so that unsteady effects such as fluctuating separation lines are averaged. The calculated flow can then be saved for later use and there is the option to visualize it in the HSV color space. This is possible by performing a conversion from cartesian to polar coordinates of the vector pair for each pixel. The resulting angle Φ describes the direction of the resulting vector and is associated with a specific color. The absolute length of the vector \vec{r} determines the pixel's brightness and is scaled to a range from 0 to 255. The saturation is set to 255 for each pixel.

The plots are created with the *streamplot* function from the matplotlib package that is based on a 2nd-order Runge-Kutta algorithm with adaptive step size. Its main inputs are a meshgrid array determining the plots size and a 2D array containing the velocity vectors. The vector field is transformed into a masked array using the previously created mask, so that *streamplot* only draws lines in the desired regions. There are several parameters that can be changed and saved by the user to adapt the plots to the individual use case. For example, the line density can be changed to modify the number of calculated streamlines. This parameter should be used with care, as the computational cost increases very quickly for higher than default values. Additionally there is an option to create a dependency between the thickness of the lines and the magnitude of the optical flow in a point. The program and its source code can be found at: <https://github.com/AeroTUBerlin/TUBflow>.

IV. Results

In this section we use our program to analyze sequences of oil-film images obtained in three different wind-tunnel tests:

- The flow over a subsonic half-diffuser;
- An incident shockwave/boundary layer interaction;
- The suction side of a high-lift wing with three propellers.

We compare the computed streamlines of the different implemented algorithms and evaluate the effect of key parameters such as the length of the image sequence and the spatial resolution.

A. Subsonic Half Diffuser Flow

For the first test case we look at the flow on the backward-facing ramp of a subsonic half-diffuser. This experiment was conducted in a closed-loop wind tunnel at a reference velocity of 20 m/s. A contour of the longitudinal velocity field on the test-section centerline together with the distribution of the wall pressure coefficient c_p and forward-flow fraction γ is provided in Fig. 2. The flow is characterized by the formation of a pressure-induced turbulent separation bubble with reverse flow near the lower ramp corner. The width of the test section is 600 mm, the length of the ramp is 337 mm and the ramp angle is 20 deg. The boundary layer upstream of the ramp is fully turbulent and has a Reynolds number based on momentum thickness of $Re_\theta \approx 5000$. Further details can be found in Weiss et al.[15].

Visualizations of the surface flow on the ramp were performed by coating the diffuser with a mixture of linseed oil, turpentine, and fluorescent pigments. Figure 3 shows a single snapshot of the oil film obtained by placing a GoPro HERO10 camera on the ceiling of the test section. The camera is set to night mode with an exposure time of 10 seconds for the snapshot image. The image reveals two prominent vortex systems near the sidewalls, along with evidence of fully attached flow on the upstream part of the ramp and downstream of the separation bubble. Near the lower corner of the ramp, the image indicates that the flow is separated, though it is difficult to clearly locate the separation and reattachment lines.

By processing a series of images of $n_{img} = 40$ at $\lambda_{scale} = 40\%$ with the three algorithms implemented in TUBflow, we can obtain the streamline visualizations that are depicted in Fig. 4. The light green contour shows the area of negative streamwise velocity as an approximation of the separated area. All algorithms are capable of capturing the major structures of the flow field with the two vortices on the ramp, reverse flow at the diffuser foot and reattachment further downstream. In contrast to the raw oilfilm visualization in Fig. 3, the separation and reattachment lines can be easily observed and critical points can be identified much easier thanks to the computed vector field. Comparing the algorithms, the Farnebäck and DIS implementation predict a similar flow field with a clear saddle point where the flow separates on the centerline, a single node at the reattachment location and smooth and slightly divergent streamlines downstream. The Horn-Schunck algorithm, on the other hand, appears to predict multiple reattachment nodes and the downstream streamlines converge into streaks that are not observable in the original image sequence. This tendency can also be found in the implementation of Liu [16], where the authors looked at a surface mounted cube and the predicted skin friction lines differ from the expected topology [17]. All algorithms struggle to some degree with identifying a

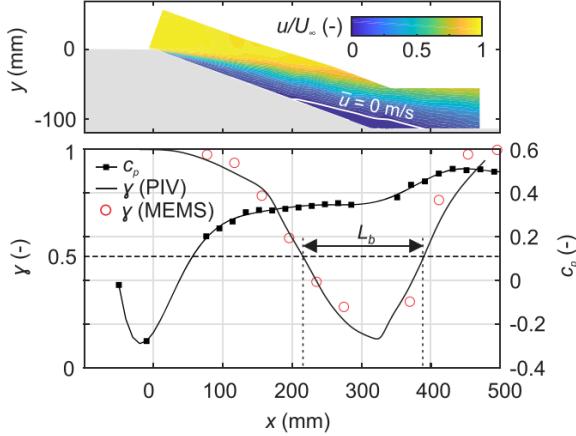


Fig. 2 Contour of longitudinal velocity field on test-section centerline (top); distributions of pressure coefficient and forward-flow fraction (bottom) [15]

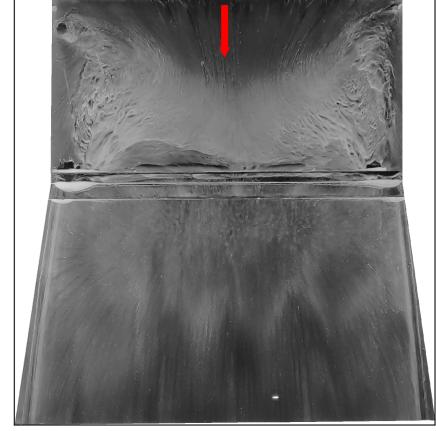


Fig. 3 Last image of the sequence of oil-film visualizations for the diffuser flow

coherent separation region. This can be attributed to accumulating oil at the edge of the ramp and a small panel gap which leads to fluctuating reflections that change the illumination of the scene and thus invalidate the constant brightness assumption of the optical flow calculation. The same explanation can be used for the bottom edge of the observed domain where oil accumulates due to a small panel gap and the algorithms struggle to predict straight streamlines as a result. Nevertheless, the output of TUBflow clearly helps the interpretation of the overall near-wall flow topology.

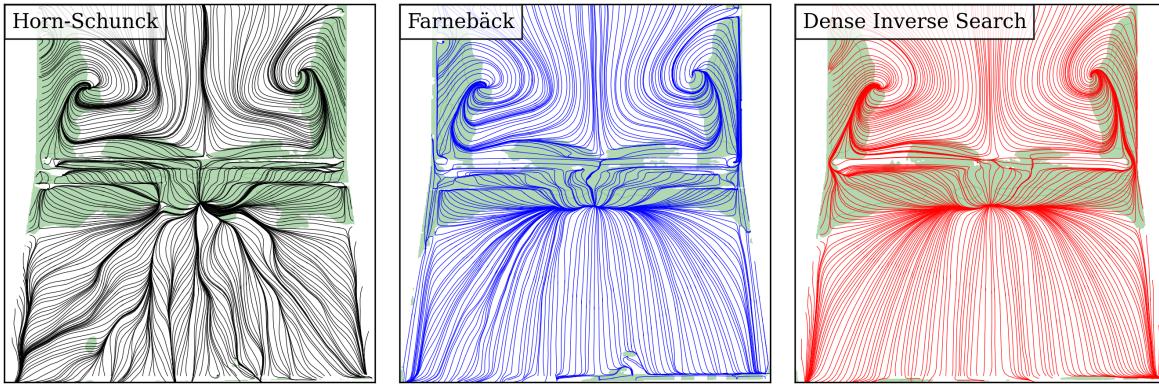


Fig. 4 Streamline visualization of the average optical flow for the different algorithms. Green contour of negative streamwise flow. $\lambda_{scale} = 40\%$, $n_{frames} = 40$

B. Turbulent Shockwave/Boundary-Layer Interaction

The second test case is a sequence of oil film images of a turbulent, incident shock-wave boundary layer interaction (SBLI) at $Ma = 2$ behind a flow deflection of $\theta = 10^\circ$. The experiments were performed in the supersonic wind tunnel at TU Berlin in a $150 \times 150 \text{ mm}^2$ test section using a non evaporating mixture of UV powder and viscous paraffin oil. The developing flow pattern was recorded using a DLSR camera through an optical access in the wind tunnel floor. Fig. 5 shows a sketch of the SBLI that develops along the centerline of the test section. It is characterized by a large separation bubble that forms due to the adverse pressure gradient of the impinging shock-wave and a high degree of unsteadiness due to the oscillating separation shock. Further details are available in the study by Rohlf et al. [19]. Fig. 6 shows the last image of the oil-flow sequence alongside the computed streamlines from TUBflow using $n_{img} = 25$ and $\lambda_{scale} = 20\%$. Note that even though λ_{scale} is lower than in the previous case, the spatial resolution of the image

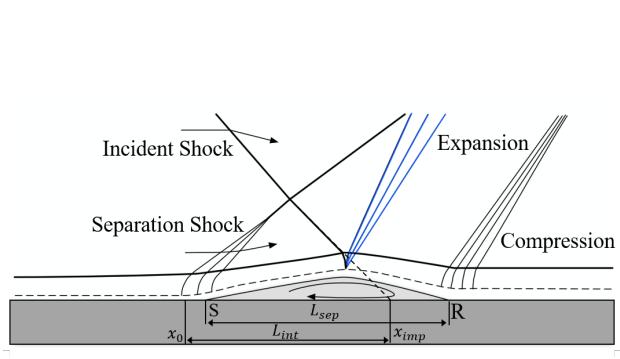


Fig. 5 Schematic structure of a strong impinging interaction with flow separation, adapted from [18]

is still much higher due to the smaller test section and higher resolution of the DLSR ($24Mpx$ vs. $\approx 8Mpx$). The streamline visualizations capture the geometric shape of the separation bubble, with a clear reattachment line and a good representation of the reverse flow region. While this also applies to the original oil-flow images, the computed velocity field would, for example, enable an automated extraction of the separation length. The DIS algorithm creates the smoothest vector field while both the Farnebäck and especially the Horn-Schunck algorithm create more artifacts and converging streamlines into streaks that do not match the perceived pattern in the original oil film image. All algorithms struggle to accurately identify the topology near the separation line which can be linked to the inherent low-frequency unsteadiness of the separation shock. Its strong movement causes a lot of oil to accumulate and flow outwards into two foci that are just visible at the edge of the image. This leads to reflections and thus invalid optical flow.

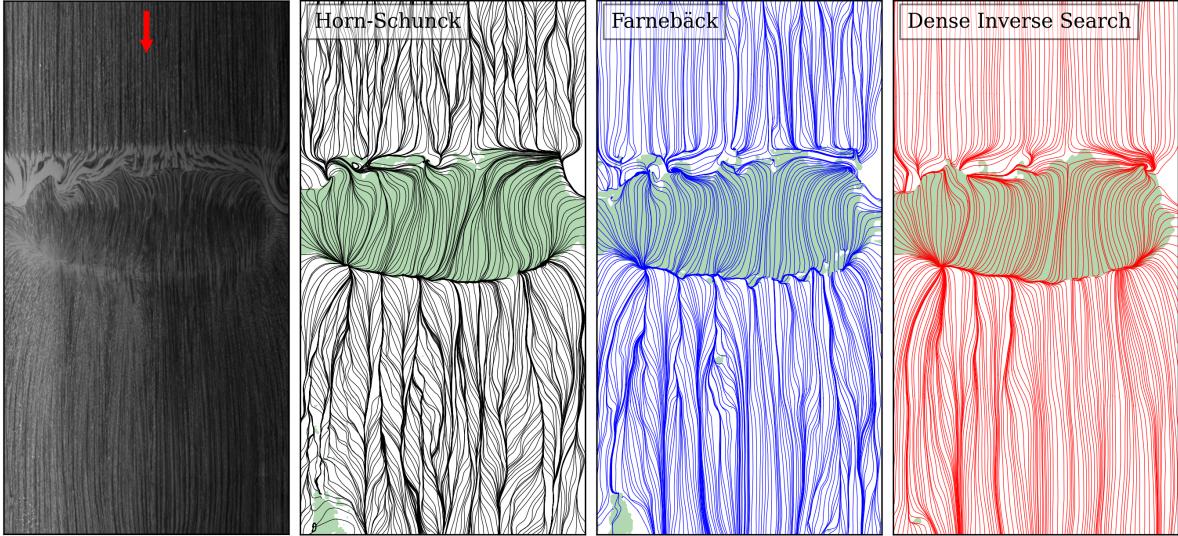


Fig. 6 Streamline visualization of the average optical flow for the different algorithms. Green contour of negative streamwise flow. $\lambda_{scale} = 20\%$, $n_{img} = 25$

Thanks to the high resolution of the image sequence, the SBLI case lends itself very well to an analysis of the influence of the scaling factor λ_{scale} . Fig. 7 shows the computed streamline visualization for the same sequence of images as in Fig. 6 for different λ_{scale} . It is quite evident that the highest resolution image does not yield the highest quality streamline visualization due to excessive artifacts. The Horn-Schunck algorithm is especially sensitive to image

noise with almost unusable results at $\lambda_{scale} > 50\%$. This is a known limitation of algorithms that globally solve the optical flow equation, which is why they are typically preceded by an image smoothing algorithm such as a Gaussian blur. On the other hand, Horn-Schunck manages to compute a reasonable reconstruction of the SBLI flow at only 5% of the original resolution (120x224 Px). The Farnebäck algorithm computes better, but still very noisy flow fields at the higher resolutions, while the representation at $\lambda_{scale} = 5\%$ is not able to capture a clear separation line. The pattern matching of the DIS algorithm is much better at dealing with image noise and manages to compute a consistent vector field over the entire range of scaling factors. The result for $\lambda_{scale} = 5\%$ is lacking some details and does not capture the real shape of the separation bubble, but would be enough for a first, quick overview of the flow topology.

To help with determining the best scaling factor for a given image sequence, the RMS of the computed average

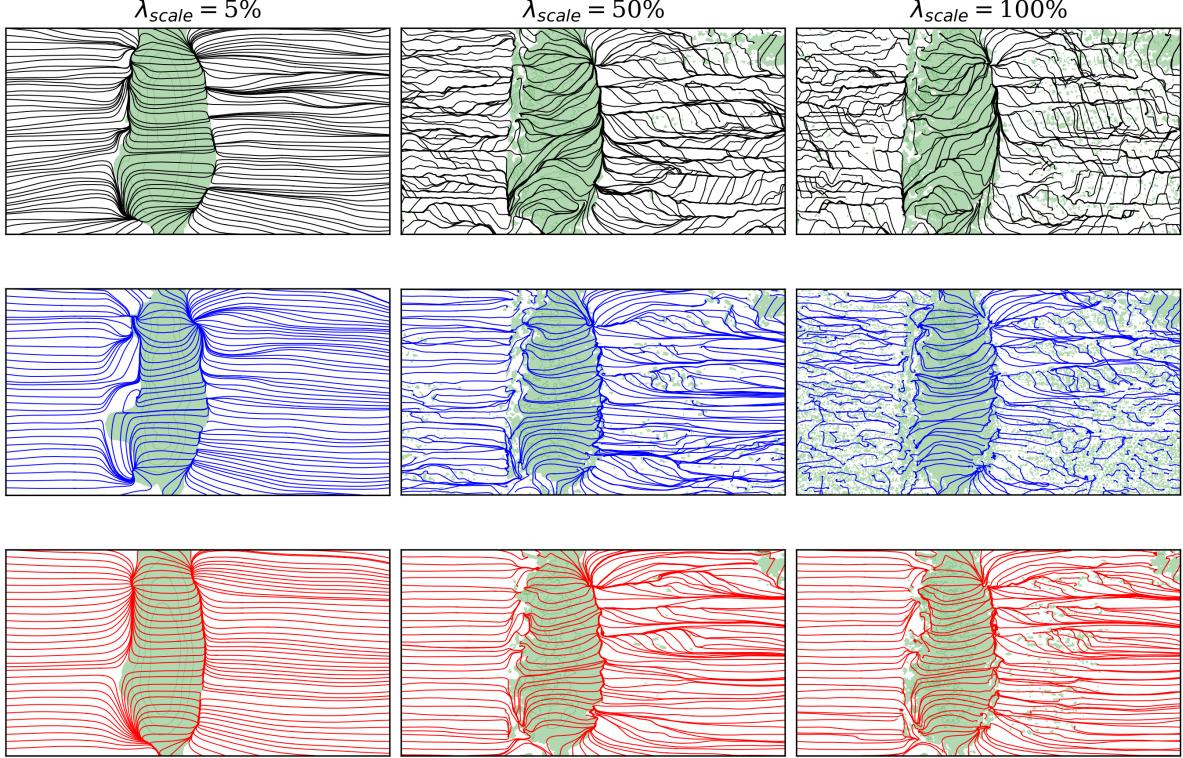


Fig. 7 Influence of the scaling factor λ_{scale} on the calculated flow field; Top: Horn-Schunck, Middle: Farnebäck, Bottom: DIS algorithm; $n_{img} = 25$

optical flow field can be plotted against the scaling factor as presented in Fig. 8. Additionally, the resulting spatial resolution is depicted. Note that the values are corrected with the scaling factor to represent the optical flow at 100% resolution. As evident, the different algorithms behave quite differently. The DIS algorithm converges to a value of $\approx 4.5px$ above $\lambda_{scale} \approx 20\%$ which indicates a similar flow field over a wide range of image scales. The Horn-Schunck and the Farnebäck algorithm, on the other hand, show a decreasing RMS for increasing λ_{scale} with very small values for higher resolutions. This indicates a convergence towards a zero velocity field resulting in a noisy, low quality flow field. In other words, the algorithms optimizer fails to find the global minimum for the optical flow equation and gets stuck in a local minimum close to the trivial solution instead. Based on this evaluation, the scaling factor of 20% was chosen for the SBLI case which arguably yields the highest quality visualization.

Another parameter worth considering is the computational cost of the optical flow analysis that is directly linked to the size of the source images. Fig. 9 displays the required time for the evaluation of the previously mentioned sequence of 25 images as a function of the number of pixels on the source image. The differences in efficiency between the algorithms can clearly be seen with the DIS algorithm requiring a maximum of $\approx 15s$ to evaluate the sequence at the highest resolution while the Horn-Schunck algorithm requires over 30 seconds to evaluate a single image pair with a total runtime of $\approx 800s$. For lower scaling factors this time does decrease considerably, but especially for longer

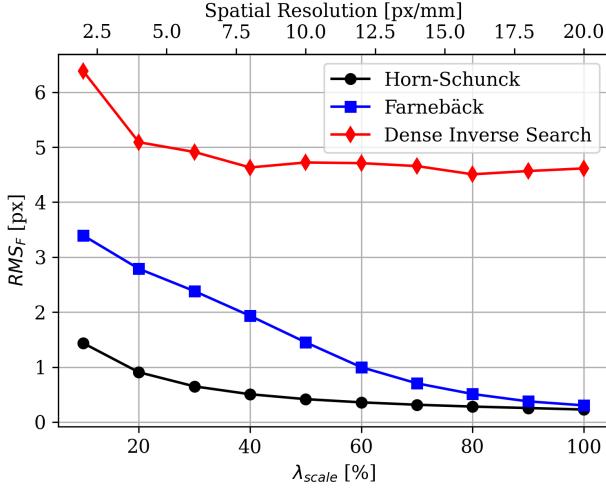


Fig. 8 RMS value of the averaged optical flow field for varying resize factors ($n_{img} = 25$)

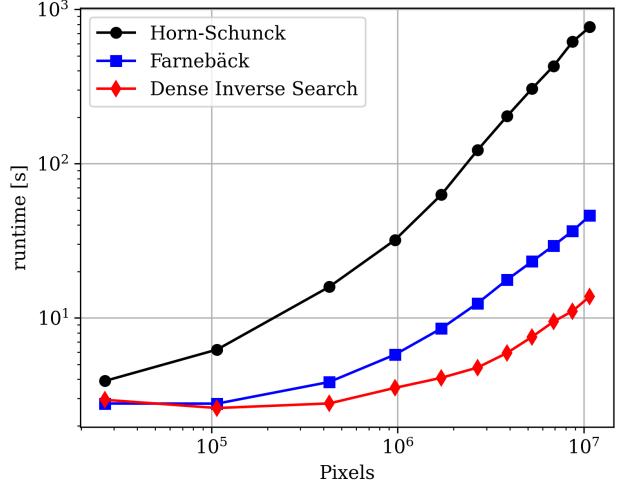


Fig. 9 Computation time

sequences the DIS or the Farnebäck algorithm are vastly more efficient.

C. High-Lift Wing with Distributed Propulsion

For the final case, we consider a high-lift wing equipped with distributed propulsion to demonstrate the applicability of our method to an industry relevant larger scale wind tunnel experiment. The wing model, as shown in Fig. 10, has a span of 1.54m with a chord length of 0.41m and is equipped with three electric propellers that each generate 65N of thrust resulting in a thrust coefficient $c_T = 0.3$. The investigations were performed at TU Berlin in the large wind tunnel facility GroWiKa with an inflow velocity of 32 m/s that resulted in a Reynolds number of $Re \approx 900,000$. Further details on the setup and the propeller configuration can be found in Gothow et al. [20].

For the flow visualization, a mixture of paraffin and linseed oil supplemented by fluorescent pigments is used. Similar

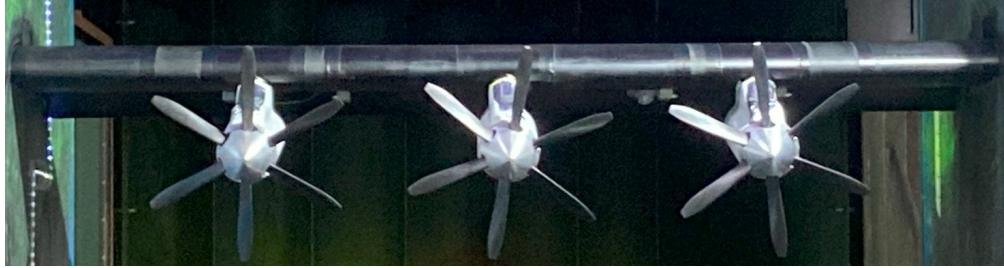


Fig. 10 High-lift wing with distributed propulsion inside the wind tunnel test section [20]

to the diffuser flow case, the oil film is recorded with a GoPro Hero 10 camera that was mounted inside the test section and was set to night mode with an exposure time of approximately 1 second for each snapshot image.

Fig. 11 shows such a snapshot as well as the streamline visualization of the average optical flow computed by TUBflow using $n_{img} = 150$ images and a scaling factor of $\lambda_{scale} = 40\%$. All three algorithms are able to capture the main flow features of this very complex flow field and the computed streamlines provide a clear representation of the surface flow topology. The effect of the propellers slipstreams that act as a boundary layer control mechanism can clearly be seen. As a result, the flow remains attached over most of the wing's upper surface, even though the angle of attack is already quite high at $\alpha = 12^\circ$. Between the propellers, there is an interaction of their wake vortices where the flow is at the edge of separation but there is no reverse flow, yet. Near the wing tips, corner separation occurs because of the finite width of the wind tunnel and the spanwise velocity component induced by the propellers. Considering the previously discussed difference in computational efficiency the results from the Farnebäck and DIS algorithm that were obtained in under 30 seconds are especially impressive.

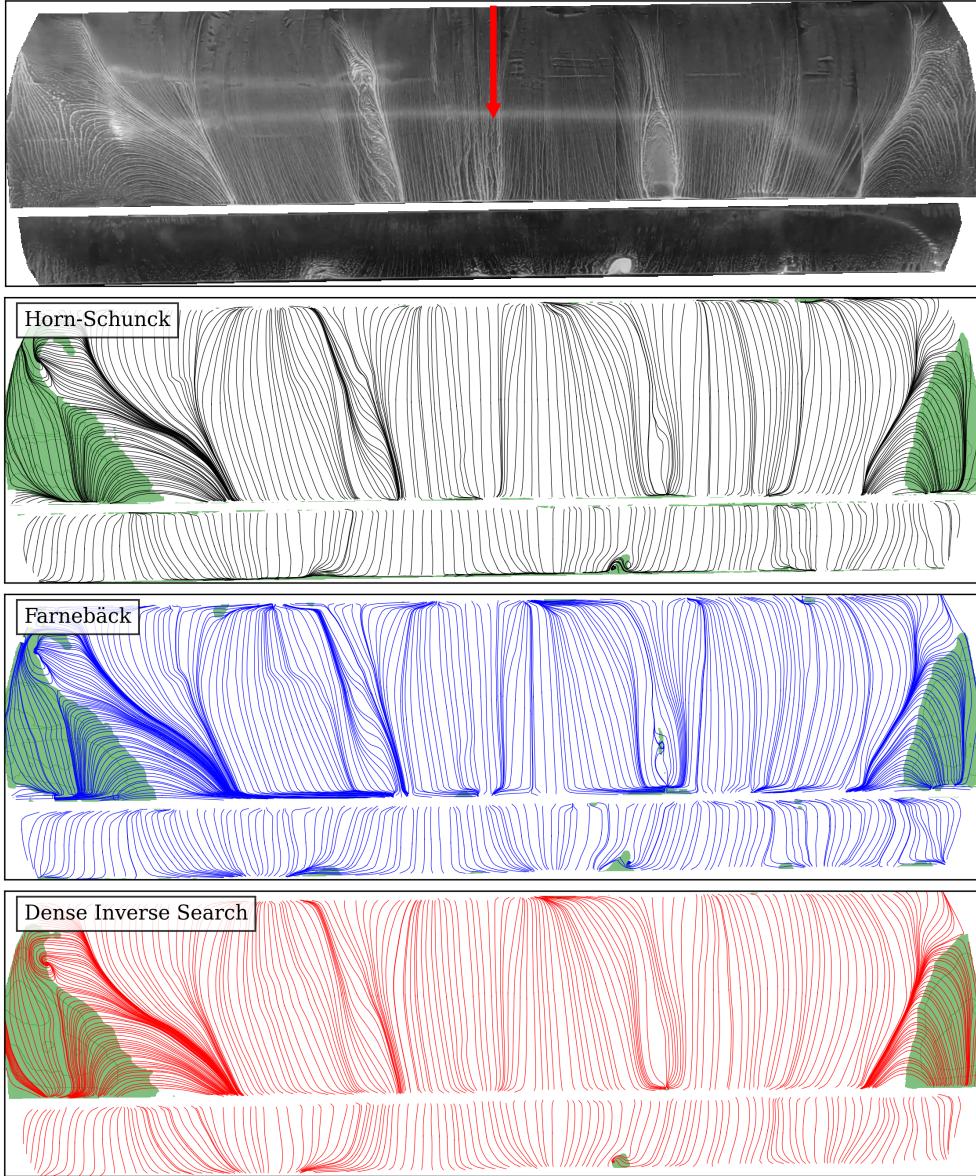


Fig. 11 Streamline visualization of the average optical flow for the different algorithms. Green contour of negative streamwise flow. $\lambda_{scale} = 40\%$, $n_{img} = 150$

Thanks to the duration of the wind tunnel runs (≈ 10 minutes) and the resulting long sequence of images, this final test case is well suited to identify the relevant parts of the recording. Fig. 12 shows the normalized standard deviation of the computed optical flow field for each of the 625 recorded images. At the start of the sequence, the wind tunnel is just starting up and the oil moves very slowly, resulting in a low standard deviation of the optical flow field. When the propellers are turned on, the oil movement increases rapidly, causing the standard deviation to shoot up and reach a peak at around 100 frames from the start. Afterwards, it tapers off as the oil dries or gets blown away by the flow. After 200 frames, the scene does not change anymore, indicating that the later images can be discarded without losing relevant information. This analysis demonstrates that the chosen subset of $n_{img} = 150$ from frame 50 to 200 is sufficient to capture the mean flow field and the rest can be safely disregarded without losing any information.

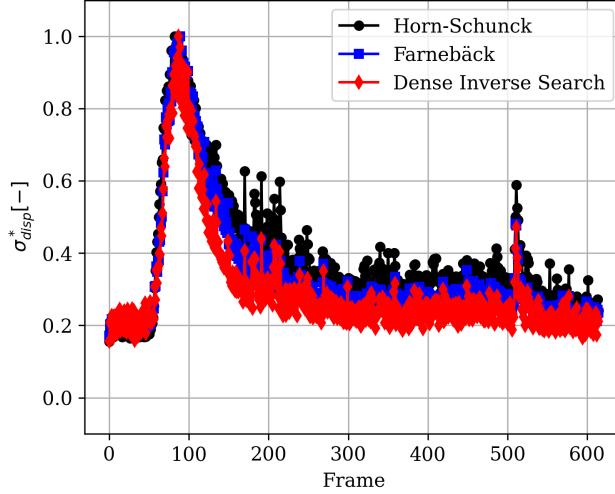


Fig. 12 Normalized standard deviation of calculated optical flow field for each image pair ($\lambda_{scale} = 40\%$)

V. Conclusion and Outlook

We have introduced TUBflow, an open-source tool with a user friendly GUI designed to digitally extract skin friction lines from a sequence of oil-film images taken in the wind tunnel. The method is based on optical flow techniques and leverages several efficient algorithms that are implemented in the Open CV library. We demonstrated the versatility of the approach by applying it to three different wind-tunnel test cases: a subsonic half-diffuser flow, an incident shock-wave/boundary-layer interaction, and the suction side of a high-lift wing with distributed propulsion.

The results show that TUBflow enhances the interpretation of oil-film visualizations by effectively capturing complex flow structures across the image sequences. In contrast to the raw oil-film images, the computed streamlines provide a much clearer representation of the surface flow topology, enabling an easier identification of flow separation and critical points. By comparing the performance of different optical flow algorithms implemented in TUBflow, we found that the Dense Inverse Search (DIS) method typically provides the best trade-off between accuracy and computational efficiency for any type of input image, while the classic Horn-Schunck algorithm works very well with low resolution source images, but struggles with noise in many cases. The analysis of key parameters such as the spatial resolution and the number of frames used for averaging the optical flow revealed several important guidelines for obtaining high-quality results. We found that the highest resolution images do not necessarily yield the best streamline visualizations due to excessive artifacts caused by image noise. Instead, a moderate downscaling of the input images using the scaling factor λ_{scale} can significantly improve the quality of the computed flow field while also reducing the computational cost. Similarly, the analysis of the standard deviation of the optical flow field over long sequences of images is very useful for selecting an appropriate subset of frames.

Since the oil-film images used in this study were acquired before TUBflow was implemented, we believe that the method should be applicable to any good quality oil-film visualization that has a reasonably stable illumination. This makes TUBflow a promising tool to aid the digital post-processing of oil-film visualizations in ground-testing campaigns. The open-source nature of the application will enable researchers and engineers to use it for their own experiments and to extend its capabilities.

References

- [1] Barlow, J. B., Rae, W. H., and Pope, A., *Low-speed wind tunnel testing*, 3rd ed., John Wiley & Sons, 1999.
- [2] Merzkirch, W., *Flow Visualization*, Elsevier Science, 1987.
- [3] Mohammed-Taifour, A., and Weiss, J., “Unsteadiness in a large turbulent separation bubble,” *Journal of Fluid Mechanics*, Vol. 799, 2016, pp. 383–412. <https://doi.org/10.1017/jfm.2016.377>.
- [4] Simmons, D., Thomas, F., Corke, T., and Hussain, F., “Experimental characterization of smooth body flow separation topography and topology on a two-dimensional geometry of finite span,” *Journal of Fluid Mechanics*, Vol. 944, 2022, p. A42. <https://doi.org/10.1017/jfm.2022.465>.

- [5] Liu, T., and Shen, L., “Fluid flow and optical flow,” *Journal of Fluid Mechanics*, Vol. 614, 2008, pp. 253–291. <https://doi.org/10.1017/S0022112008003273>.
- [6] Liu, T., Montefort, J., Woodiga, S., Merati, P., and Shen, L., “Global Luminescent Oil-Film Skin-Friction Meter,” *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 476–485. <https://doi.org/10.2514/1.32219>.
- [7] Horn, B. K. P., and Schunck, B. G., “Determining optical flow,” *Artificial Intelligence*, Vol. 17, No. 1-3, 1981, pp. 185–203. [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2).
- [8] Liu, T., “Global skin friction measurements and interpretation,” *Progress in Aerospace Sciences*, Vol. 111, 2019, p. 100584. <https://doi.org/10.1016/j.paerosci.2019.100584>.
- [9] Schmidt, B. E., and Sutton, J. A., “High-resolution velocimetry from tracer particle fields using a wavelet-based optical flow method,” *Experiments in Fluids*, Vol. 60, No. 3, 2019, p. 37. <https://doi.org/10.1007/s00348-019-2685-6>.
- [10] Corpetti, T., Heitz, D., Arroyo, G., Mémin, E., and Santa-Cruz, A., “Fluid experimental flow estimation based on an optical-flow scheme,” *Experiments in Fluids*, Vol. 40, No. 1, 2006, pp. 80–97. <https://doi.org/10.1007/s00348-005-0048-y>.
- [11] Schmidt, B. E., and Woike, M. R., “Wavelet-Based Optical Flow Analysis for Background-Oriented Schlieren Image Processing,” *AIAA Journal*, Vol. 59, No. 8, 2021, pp. 1–8. <https://doi.org/10.2514/1.J060218>.
- [12] Farnebäck, G., “Two-Frame Motion Estimation Based on Polynomial Expansion,” *Lecture Notes in Computer Science*, Vol. 2749, Springer, Berlin, Heidelberg, 2003, pp. 363–370. https://doi.org/10.1007/3-540-45103-X_50.
- [13] Kroeger, T., Timofte, R., Dai, D., and Van Gool, L., “Fast Optical Flow Using Dense Inverse Search,” *Computer Vision – ECCV 2016*, edited by B. Leibe, J. Matas, N. Sebe, and M. Welling, Springer International Publishing, Cham, 2016, pp. 471–488. https://doi.org/10.1007/978-3-319-46493-0_29.
- [14] Lucas, B. D., and Kanade, T., “An iterative image registration technique with an application to stereo vision,” *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1981, pp. 674–679. <https://doi.org/10.5555/1623264.1623280>.
- [15] Weiss, J., Steinfurth, B., Chamard, L., Giani, A., and Combette, P., “Spectral Proper Orthogonal Decomposition of Unsteady Wall Shear Stress Under a Turbulent Separation Bubble,” *AIAA Journal*, Vol. 60, No. 4, 2022, pp. 2150–2159. <https://doi.org/10.2514/1.J061087>.
- [16] Liu, T., “Extraction of skin-friction fields from surface flow visualizations as an inverse problem,” *Measurement Science and Technology*, Vol. 24, No. 12, 2013, p. 124004. <https://doi.org/10.1088/0957-0233/24/12/124004>.
- [17] Martinuzzi, R., and Tropea, C., “The Flow Around Surface-Mounted, Prismatic Obstacles Placed in a Fully Developed Channel Flow (Data Bank Contribution),” *Journal of Fluids Engineering*, Vol. 115, No. 1, 1993, pp. 85–92. <https://doi.org/10.1115/1.2910118>.
- [18] Babinsky, H., and Harvey, J. K., *Shock Wave-Boundary-Layer Interactions*, Cambridge University Press, 2011. <https://doi.org/10.1017/CBO9780511842757>.
- [19] Rohlf, L., Stab, I., and Weiss, J., “Experimental Investigations of Incident Shockwave Boundary Layer Interactions in a Continuously Operating Supersonic Wind Tunnel,” *AIAA AVIATION 2022 Forum*, American Institute of Aeronautics and Astronautics, Chicago, IL, 2022. <https://doi.org/10.2514/6.2022-4134>.
- [20] Gothow, A., Weiss, J., Bardenhagen, A., Bergmann, D. P., Denzel, J., and Strohmayer, A., “Experimental Parameter Study of Distributed Electric Propulsion on a 2D Wing Model in High-Lift Configuration,” *AIAA AVIATION 2023 Forum*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2023. <https://doi.org/10.2514/6.2023-3540>.