Calibration of ILIDS and PDPA droplet sizing systems

by

Sebastian Kosch

A thesis submitted in conformity with the requirements for the degree of
Master of Applied Science

Graduate Department of the Department of Mechanical and Industrial Engineering,
University of Toronto

Sebastian Kosch: *Calibration of ILIDS and PDPA droplet sizing systems.*
Master's thesis (MASc), Department of Mechanical and Industrial Engineering,
University of Toronto.
April 2015.

Abstract

Laser-based droplet sizing techniques are widely used in industrial prototyping and diagnostics applications. However, their ability to immediately output highly precise droplet size distribution charts, seemingly without any significant user input, belies their vulnerability to bad input parameters, noise, and optical aberrations that can only be identified and resolved by means of careful calibration using monodisperse droplets. The contribution of this paper is threefold: first, we provide a review of well-known and obscure sticking points identified in various publications and how they apply in the context of ILIDS and PDPA calibration procedures; second, we present a novel, cheap and simple droplet generator design of the continuous-stream type, based on the rotary actuator of a computer hard drive; third, we submit a proof-of-concept design of a computer vision algorithm to eliminate the need for camera calibration in combined PIV/ILIDS systems.

Acknowledgements

I must thank, first and foremost, Prof. Nasser Ashgriz, who has supported me throughout this thesis (and on projects beyond) with his patience, experience, and knowledge while allowing me the room to work in my own way. Thank you for entrusting me, a Civil Engineering major, with your research project—and for indulging all of my crazy ideas.

I am hugely indebted to our technologists, machinists, and lab wizards Terry Zak, Osmond Sargeant, and Keith Hockley for showing me the ropes. Without your guidance and advice, this thesis could not have been written. The same respect is due to Amirreza Amighi, who guided me through the early stages of my work and imbued me with an appreciation for well-designed experiments.

Finally I want to express my gratitude to my family and my best friend, QP, for everything. I wouldn't have made it here without you guys.

Contents

List of Tables

List of Figures

Introduction

Fluid sprays find application in numerous fields, ranging from rocket propellant atomizers [3, 4] to plasma coating systems [5] and from medical inhalers [6] to inkjet printers [7]. More often than not, the droplet size is a major factor in how well these systems perform. Consequently, the ability to measure the characteristics of a given spray to a certain degree of accuracy is a crucial requirement during the design process of many products.

Due to the behaviour and size of the droplets in typical sprays, only non-intrusive methods can reliably report droplet characteristics. The most popular techniques rely on illumination of the spray with laser light. The droplets scatter the light, and it is picked up by a set of detectors, such as photomultiplier tubes or digital cameras. Depending on the position of the detectors, the motion of the droplets, their size, and the laser light's known properties, the detected patterns can be processed to yield information about the spray's characteristics.

Two systems are particularly popular: ILIDS (Interferometric Laser Imaging for Droplet Sizing), which relies on imaging an illuminated section of the spray, and PDPA (Phase-Doppler Particle Analysis), which makes a series of point measurements at the intersection of multiple laser beams. The former, in combination with a PIV system (Particle Image Velocimetry) for velocity measurements, promises a simple way of obtaining flow and size fields for entire areas of the spray plume at a time. The latter exploited promises very high precision measurements of both velocity and size at single points in the spray.

Although both types of systems have been commercially available for decades, the scientific publications cover mainly variations and improvements on previous designs as well as novel applications of the techniques. Literature dealing with the practical aspects of their use is scant. In particular, we felt a need for a high-level guide to calibrating the devices using standard lab equipment, including advice on generating uniform droplets and a review of potential error sources.

The objective of the presented work was threefold:

— to calibrate our Dantec IPI (ILIDS) droplet sizing system and to verify its accuracy photographically;
— to identify and remedy error sources inherent in the ILIDS technique;
— to understand, document, and verify the algorithm underlying the calibration procedure in the TSI FlowSizer PDPA system.

The need to verify the systems' accuracy was prompted by discrepancies found by previous students between ILIDS and PDPA results for identical spray nozzles.

The main contributions made towards these ends are as follows:

— a novel droplet generator design, which resulted from extensive experimental tests to identify the ideal source of monodisperse droplets for calibration purposes;
— a computer-vision-based technique that eliminates the need for camera calibration in combined ILIDS/PIV systems, which addresses the centre discrepancy effect;
— a literature review of potential error sources in PDPA measurements.

1.1 Organization of this thesis

Chapter Two contains a high-level review of the light scattering models underlying the sizing technologies.

Chapter Three offers an overview over existing droplet generation techniques and introduces the novel harddrive-based droplet generator design.

Chapter Four provides an introduction to the ILIDS method, along with a summary of theoretical and practical limitations. We focus on different setups and how to avoid common pitfalls—including in particular a proposed homography estimation algorithm to overcome the problem of centre discrepancies. We also provide a discussion on and example of the calibration process.

Chapter Five deals with the common sticking points encountered in PDPA, and offers an explanation of the inner workings of the TSI FlowSizer software that will help researchers understand the calibration process.

Chapter Six contains a summary of the contributions and an outlook on potential future work.

Appendices containing code and data are provided for completeness.

Geometrical optics and Mie scattering

In this chapter, we will discuss some of the fundamentals of Mie scattering. The majority of the equations and derivations herein are adapted directly from Albrecht et al. [1]. For the purposes of this discussion, we will model light as a coupled pair of electrical and magnetic fields, **E** and **B**, which at every point in space obey Maxwell's equations

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \tag{2.1}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2.2}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2.3}$$

$$\nabla \times \mathbf{B} = \mu_0 \sigma \mathbf{E} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}, \tag{2.4}$$

where the constants ε_0 and μ_0 represent the electric permittivity and magnetic permeability in vacuum, respectively; ρ is the local charge density, and σ is the electric conductivity (such that $\sigma\mathbf{E} = \mathbf{J}$ is the current density).

Electromagnetic radiation propagates in a direction **S** orthogonal to the planes of oscillation of **E** and **B**, which are themselves orthogonal to one antoher:

$$\mathbf{S} = \mathbf{E} \times \mathbf{B}. \tag{2.5}$$

S is called the *Poynting vector*.

Note that this definition leaves **S** invariant to a rotation of **E** and **B** around **S**. The rotation angle is called *polarization* and need not be constant along the line of propagation ("circular polarization"), although we will here assume that it is ("linear polarization") to simplify our derivations. This assumption is justified in typical spray characterization applications, for the *Brewster windows* employed as mirrors in the commonly used gas lasers act as polarization filters.

We will also assume here that **S** is approximately constant over the surface of the droplet. In other words, we assume the incoming light wave to be plane. Whether this is a reasonable assumption depends on the configuration of the laser optics, on the width of the laser beams and on the size of the droplet. We will see in Section 2.3 that more complex models without this assumption can be used, but they are not suitable for showcasing the fundamental concepts behind droplet light scattering. Finally, we assume the droplet in question to be perfectly spherical.

When light hits a spherical, transparent droplet, three phenomena can be observed:

1. Diffraction
2. Reflection
3. Refraction

We will consider each one and derive its influence on the scattered field.

In the following discussion, we will use a coordinate system centered on the droplet. Let the z-axis be aligned with the direction of the plane wave incident on the droplet, and define the rotation of the x-y-plane to be at an arbitrary but fixed angle with respect to light source and detector. Consider then an infinitesimally small light ray scattered by the droplet. We shall designate the direction of the scattered ray to be at a deviation angle φ with respect to the x-axis, and to be at a deviation angle θ with respect to the z-axis.

We also define the plane between the scattered direction and the scattered direction's projection onto the x-y-plane as the *scattering plane*. This allows us to treat any incident polarization as a linear combination of polarization parallel to that scattering plane (θ-field) and perpendicular to it (φ-field).

2.1 Diffraction

Diffraction, sometimes described as the "bending" of light waves, is the phenomenon observed when plane waves—even material waves – encounter the edge of an obstacle and

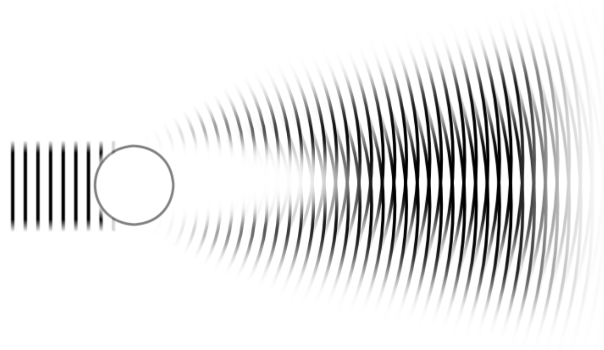


Figure 2.1: Diffraction

spread out spherically behind it. While most of the scattered intensity is focused along the original wave path, the scattered fields from two or more edges will interfere behind the obstacle and create an intensity pattern dependent on both the wavelength and the edges' separation.

A droplet in a plane wave, just like a hole of the same size, has one circular edge (or, in Huygensian terms, a set of infinitely many point-sized edges arranged in a circle). The diffracted fields interfere behind the particle, as shown in Figure 2.1, to form a periodic intensity pattern.

The viability of exploiting diffraction as a spray characterization principle is limited, because the phenomenon is strongest when droplet sizes are close to the light's wavelength. Nevertheless, diffraction-based droplet sizing systems are highly popular, particularly because they require no calibration.

We will thus here on scattering phenomena relevant to techniques such as *Phase Doppler Anemometry* and *Interferometric Particle Imaging*.

2.2 Reflection and refraction

Detectors used in spray diagnostics are ultimately based on the photoelectric effect, and thus sensitive only to incident light intensity. This is the case both for the CMOS and CCD cameras used in interferometric techniques and for the photovoltaic cascade amplifiers commonly used in Phase Doppler setups. As a result, we are interested in accurately modelling the light intensity at a location $\langle \varphi, \theta, z \rangle$ on the detector surface.

A general description of the scattering process would trace a ray of light through the droplet. At every encounter with an interface (air→water, or water→air from inside the droplet), some of the light is *reflected off* the interface and some is *refracted through* the interface. When first incident, the refracted light ray is cast *into* the droplet; subsequent scattering events will refract some light *out* of the droplet while the remainder continues being reflected within the droplet until all energy is refracted out or dissipated as heat. Scattering events are enumerated; the *scattering order N* refers to the N^{th} reflection/refraction.

Total intensity at any location is a function of the intensity of all infinitesimal light rays scattered onto that location, but also of the phases of those light rays, as two equally intense light rays will extinguish one another if they are in opposite phase. We will thus derive expressions for both intensity and phase of infinitesimal light rays scattered through the droplet.

2.2.1 Intensity from interface separation

Consider an infinitesimal ray incident, at an angle ϑ_i, on the surface of the droplet. We are interested in the respective proportions of light reflected and refracted.

Following the continuity requirement for components of **E** and **B** parallel and perpendicular to the droplet's interface (depending on the polarization component), and the conservation of energy, we can arrive at *Fresnel equations:*

$$r_\theta = \frac{m\cos\vartheta_i - \cos\vartheta_t}{m\cos\vartheta_i + \cos\vartheta_t}, \quad r_\varphi = \frac{\cos\vartheta_i - m\cos\vartheta_t}{\cos\vartheta_i + m\cos\vartheta_t} \tag{2.6}$$

with $\vartheta_t = \arcsin\left(\frac{\sin\vartheta_i}{m}\right)$, and m the relative refractive index of droplet and medium (e.g. $m = 1.33/1.0$ for the water→air transition).

Squaring these numbers yields a reflectance coefficient $R = |r|^2$ for both polarization directions, and from $R + T = 1$ follows the value of its complement, the transmission coefficient T.

2.2.2 Intensity from geometry

An infinitesimal plane wave ray reflected off of a curved surface will inevitably experience deviation into a diverging ray (convex surface) or a converging ray (concave surface).

Consider Figure 2.2. An annular ray of surface area dA_i is incident on the droplet of

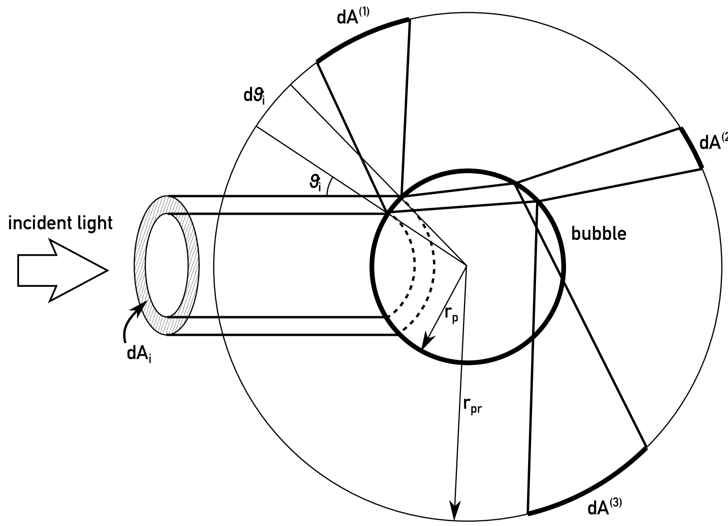


Figure 2.2: Ray divergence due to curved interfaces. Note that here a bubble in water is used instead of a droplet. Adapted from Albrecht et al. [1]

radius r_p. The first-order scattering is shown: the infinitesimal thickness of the ray results in an annular illuminated surface region between the angles ϑ_i and $\vartheta_i + \mathrm{d}\vartheta_i$ of area

$$\mathrm{d}A_i = \pi r_p^2 \sin(2\vartheta_i + \mathrm{d}\vartheta_i)\sin(\mathrm{d}\vartheta_i) \tag{2.7}$$

The reflected portion of the light diverges; its cross-section has a divergence angle of $2\,\mathrm{d}\vartheta_i$.

We can now imagine a sphere around the droplet. Its radius equals the distance to the detector, r_{pr}. The area of that sphere illuminated by the reflected portion of $\mathrm{d}A_i$ can be geometrically shown to be

$$\mathrm{d}A^{(1)} = 4\pi r_{pr}^2 \sin(2\vartheta_i + \mathrm{d}\vartheta_i)\sin(\mathrm{d}\vartheta_i). \tag{2.8}$$

The ratio between them is then

$$I_r^{(1)} = \frac{\mathrm{d}A_i}{\mathrm{d}A^{(1)}} = I_w \frac{r_p^2}{4r_{pr}^2}. \tag{2.9}$$

The ratio r_p^2/r_{rp}^2 varies based on the setup; to normalize it we place a point source of the same intensity of light at the centre of the droplet and consider its intensity at r_p^2:

$$I_P = I_w \frac{r_p^2}{r_{pr}^2} \tag{2.10}$$

We drop this term from equation (2.9) and arrive at a fixed fraction, or *gain factor*, of $G^{(1)} = \frac{1}{4}$. This number represents the "dilution of energy" associated with the divergence inherent in reflection off a curved surface.

Similar gain factors can be derived geometrically for any other scattering order N of light leaving the droplet:

$$G^{(N)} = \left| \frac{\sin\vartheta_i \cos\vartheta_i}{\sin D^{(N)}} \left(2\frac{(N-1)\cos\vartheta_i}{m\cos\vartheta_t} - 2 \right)^{-1} \right| \tag{2.11}$$

with

$$D^{(N)} = 2(N-1)\arcsin\left(\frac{\sin\vartheta_i}{m}\right) - (N-2)\pi - 2\vartheta_i. \tag{2.12}$$

2.2.3 Path-length differences

Whether light rays interfere constructively or destructively depends on the phase difference between them. Since all rays are generated in phase in the laser cavity, possible phase differences must be due to path length differences, distortion of the wave (focal shift) or reflection effects. We shall first discuss the treatment of path lengths.

To effectively compare path lengths between various rays, it is practical to establish a reference path length for common comparison. This reference ray is typically pictured as passing through the center of the droplet—an optical anomaly as it were, but geometrically useful.

A ray incident at an angle ϑ_i and reflected off the surface of the particle then has a path length $2kr_p \cos\vartheta_i$ shorter than that of the reference ray, where k is the wave number of the light ray. This situation is illustrated in Figure 2.3a.

A more general equation for phase length difference is

$$\phi^{(N)} = 2kr_p(\cos\vartheta_i - (N-1)m\cos\vartheta_i), \tag{2.13}$$

where ϕ stands for the phase difference and is not to be confused with the scattering plane's orientation φ.

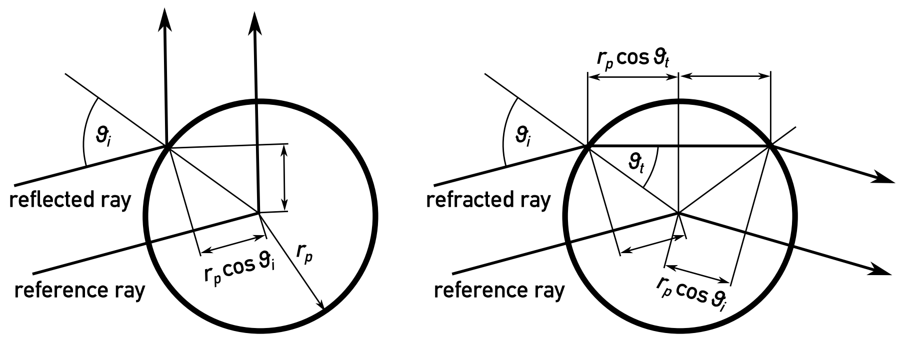


Figure 2.3: Path length differences, adapted from Albrecht et al. [1]

2.2.4 Phase shifts at reflection

A ray's phase is further affected upon reflection. This relationship is quite complex and depends on the critical and Brewster angles between the two media, the polarization angle and the relative refractive index. For example, the phase is flipped when a ray is reflected off a medium with higher refractive index (such as a water droplet), if the ray is polarized parallel to the interface and the incident angle does not exceed the Brewster angle. Similar relationships hold for all other conditions.

The derivations of these phase change formulae follow from the Fresnel equations (2.6) shown above and from the understanding that incident and refracted amplitudes must always sum to the refracted amplitude (which is zero for total reflection), i.e. continuity of amplitudes across interfaces. They are widely published (e.g. Hecht [8]) and algebraically straightforward but lengthy, and thus not included here.

2.2.5 Phase shifts through focussing

While the phase of a plane wave ray does not change (except for the wave's propagation through time and space), the phase of a diverging or converging ray is affected by the angle determining the ray width change as follows:

$$\phi^{(N)} = \pi \left((N-1) - \frac{1}{2} \left[1 + \operatorname{sgn} \left(\frac{\partial D^{(N)}}{\partial \vartheta} \right) \right] \right), \tag{2.14}$$

The derivation given in van de Hulst [9] explains this as a phase shift of $\pi/2$ whenever an astigmatic ray passes a focal line.

2.2.6 Scattering functions

The expressions given in the above sections can be combined into *scattering functions* for both polarization directions as follows:

$$S_1^{(N)} = \sqrt{i_\varphi^{(N)}}\exp(j\phi_\varphi^{(N)}), \quad S_2 = \sqrt{i_\theta^{(N)}}\exp(j\phi_\theta^{(N)}). \tag{2.15}$$

Here, i is the intensity coefficient

$$i_{\theta,\varphi}^{(N)}(\lambda, d_p, m, \vartheta_i) = \left(\frac{\pi d_p}{\lambda}\right)^2 \alpha_{\theta,\varphi}^{(N)} G^{(N)} \tag{2.16}$$

defined in terms of wavelength λ, droplet diameter d_p, gain factor $G^{(N)}$ as given in (2.11) above, and $\alpha^{(N)}$, which is a shorthand for the product of all applicable reflectance and transmittance coefficients:

$$\alpha_{\theta,\varphi}^{(N)} = \begin{cases} R_{\theta,\varphi}, & N = 1 \\ \\ R_{\theta,\varphi}^{N-2} T_{\theta,\varphi}^2, & N \geq 2 \end{cases}. \tag{2.17}$$

The $\exp(j\phi_{\varphi,\theta}^{(N)})$ terms represent the phase shift added to the wave function term in (2.18)

We can use these two-component scattering functions to write a full expression relating the incident light wave $\mathbf{E}_i$ in its components $\langle E_{ix}, E_{iy}\rangle$ to the wave received by the detector, $\mathbf{E}_r$:

$$\begin{aligned} \mathbf{E}_r = & \frac{\exp(-jkr_{pr})}{kr_{pr}} \begin{bmatrix} \cos\beta_{\theta x} & \cos\beta_{\varphi x} \\ \cos\beta_{\theta y} & \cos\beta_{\varphi y} \end{bmatrix} \\ & \begin{bmatrix} \sum_{N=0}^{\infty} S_1^{(N)} & 0 \\ 0 & \sum_{N=0}^{\infty} S_2^{(N)} \end{bmatrix} \times \begin{bmatrix} -\sin\varphi & \cos\varphi \\ \cos\varphi & \sin\varphi \end{bmatrix} \begin{bmatrix} E_{ix} \\ E_{iy} \end{bmatrix} \end{aligned} \tag{2.18}$$

The first term evaluates to a sinusoid function corresponding to the scalar component of the light wave. The second term (and first matrix) represents the geometrical relationship between the light source and the detector, expressed in terms of the off-angle β. Similarly, the matrix in φ represents the angle between light source and droplet. By multiplying the β matrix with the appropriate scattering functions (which are summed over all scattering orders N), we arrive at the correct intensity at the detector's location.

2.3 Mie theory

The above derivations, commonly termed *geometrical optics*, are physically intuitive and therefore useful in the understanding and design of measurement instrumentation. Geometrical optics, however, are limited to incident waves that are both plane and much smaller in wavelength than the droplet diameter. A more general approach is therefore desirable for carrying out detailed numerical simulations.

Such a numerical system was developed independently by Ludvig Lorenz and Gustav Mie based on analytical solutions presented here briefly. The Mie scattering theory decomposes the incoming light into an infinite series of not plane, but spherical waves concentrical with the droplet. The spherical waves are not uniform; their amplitude varies with the angle ϑ.

The derivation in scalar terms assumes a scalar field magnitude ψ that solves

$$\nabla^2\psi + k_m^2\psi = 0 \tag{2.19}$$

where k_m^2 is the magnitude of the field in the surrounding medium.

In spherical coordinates, this leads to

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial\psi}{\partial r}) + \frac{1}{r^2\sin\vartheta}\frac{\partial}{\partial\vartheta}(\sin\vartheta\frac{\partial\psi}{\partial\vartheta}) + \frac{1}{r^2\sin^2\vartheta}\frac{\partial^2\psi}{\partial\psi^2} + k_m^2\psi = 0 \tag{2.20}$$

Which is solved by

$$\psi(r,\vartheta,\varphi) = R(r)\Theta(\vartheta)\exp(\pm im\varphi) \tag{2.21}$$

according to Ng [10].

Solving for the function governing angular dependence, $\Theta(\vartheta)$, leads to Legendre polynomials. Polar plots of the first five such functions of both types are shown in Figure 2.4.

Solving for $R(r)$, the function describing the intensity of the spherical waves as it drops with distance from the droplet, leads to vector spherical harmonics. Two-dimensional plots of a few orders of such "Riccati-Bessel functions" are shown in Figure 2.5.

Each partial wave, in turn, represents the sum of scattering effects up to some scattering

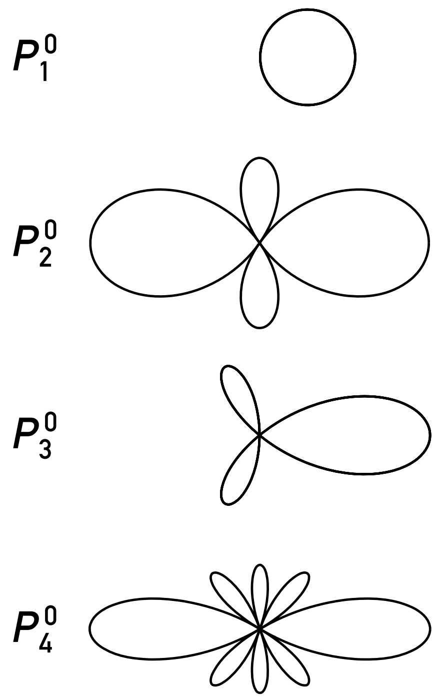


Figure 2.4: A few associated Legendre polynomials $\Theta(\vartheta)$ of order zero (here labelled *P*). Incident light is coming from the left.

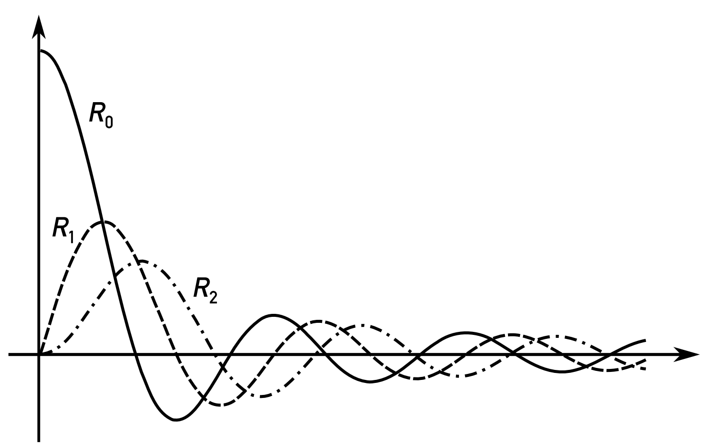


Figure 2.5: Some Riccati-Bessel functions governing the radial decay of the partial waves.

order N (in practice, no infinite sum can be taken; the maximum scattering order to achieve negligible errors depends on the relative wavelength). With every refraction out of the droplet—thus adding to the total external field—the *internal* partial wave loses intensity according to respective reflectance and transmittance values.

The complete model, then, provides a semi-closed expression for the scattering functions S_1 and S_2:

$$S_1(\theta) = \sum_{n=1}^{\infty} a_n \pi_n(\theta) + b_n \tau_n(\theta) \tag{2.22}$$

$$S_2(\theta) = \sum_{n=1}^{\infty} a_n \tau_n(\theta) + b_n \pi_n(\theta) \tag{2.23}$$

$$a_n = \frac{2n+1}{2n(n+1)}(1 - R_{a_n}^{MM} - \sum_{p=2}^{\infty} T_{a_n}^{MP}(R_{a_n}^{PP})^{p-2} T_{a_n}^{PM}) \tag{2.24}$$

$$b_n = \frac{2n+1}{2n(n+1)}(1 - R_{b_n}^{MM} - \sum_{p=2}^{\infty} T_{b_n}^{MP}(R_{b_n}^{PP})^{p-2} T_{b_n}^{PM}) \tag{2.25}$$

$$\text{e.g. } R_{b_n}^{PP} = \frac{\zeta_n(m\frac{\pi d_p}{\lambda})\zeta_n'(\frac{\pi d_p}{\lambda}) - m\zeta_n(\frac{\pi d_p}{\lambda})\zeta_n'(m\frac{\pi d_p}{\lambda})}{-\frac{\pi d_p}{\lambda}\xi_n(m\frac{\pi d_p}{\lambda})\zeta'(\frac{\pi d_p}{\lambda}) + m\zeta_n(\frac{\pi d_p}{\lambda})\frac{\pi d_p}{\lambda}\xi'(m\frac{\pi d_p}{\lambda})} \tag{2.26}$$

where *PP* stands for interal reflection, *MM* external reflection, and *MP* and *PM* for refraction into and out of the particle, respectively. The expression for $R_{b_n}^{PP}$ is shown above to illustrate the coefficents' form. Note that in the above equations, π and τ are associated Legendre functions and ζ and ξ are Riccati-Bessel functions.

Some numerical results are presented in Fig. 2.6 below. The same methods were used in papers like Naqwi and Menon [11] to predict the angular variation of scattered intensity that is crucial in designing phase-Doppler systems.

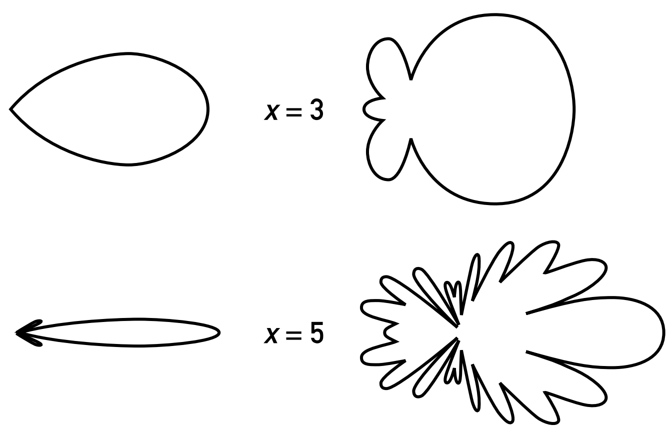


Figure 2.6: Scattered light intensity based on light incident from the left on a spherical particle of size parameters $x = 3$ and $x = 5$, where $x = \frac{\pi D_d m}{\lambda}$. Only perpendicularly polarized light intensities are shown. Right column shows logarithmically scaled intensities. Figure adapted from Albrecht et al. [1].

Monodisperse droplet generation

To calibrate any droplet sizing device, we need droplets of known and uniform size. Sprays or streams of such uniform droplets are called *monodisperse*, and many different varying approaches to generating them have been proposed, each one with advantages and drawbacks.

The most basic type of droplet generator is a capillary tube, for instance a hypodermic needle or a pulled glass pipette. Droplets are generated as the liquid flows through the tube due to its own weight. As the liquid leaves the tube, it wets the tip of the tube and forms a bead held together by surface tension. Eventually, the bead's gravitational forces overcome the attraction to the tube surface, and the drop separates from the tube.

Given a liquid and its physical properties, the only remaining controllable variable is the diameter of the capillary tube tip. As a rule, droplets generated in this fashion will be significantly larger than the tube diameter from which they grow. Most droplet generators are designed to prevent this from happening:

— *Aerodynamic* droplet generators use coaxial air flow to shear the forming droplet off of the capillary tip before it can grow to full size.

— *On-demand* droplet generators use a pressure pulse to eject a fixed amount of liquid out of the capillary (or other orifice).

— *Continuous-stream* droplet generators use mechanical vibrations to break up a continuous jet of liquid emanating from the capillary into monodisperse droplets.

More exotic types of droplet generators exist: Walton and Prewett [12] suggested that water falling on spinning disks is propelled outwards, forming nearly monodisperse droplets, and several improved designs have been published since. Another approach, e.g. used by Merritt and Drinkwater [13], involves mechanized dipping of a needle into a liquid reservoir, and then flicking it so as to produce one droplet.

3.1 Aerodynamic droplet generators

Aerodynamic droplet generators use a stream of gas to assist the droplet in detaching from a needle, orifice or stylus before it grows large enough to do so by force of gravity alone. The technique promises a simple way to produce small droplets, but it can be difficult to control the exact size of the droplets.

Allan et al. [14] provide a history of aerodynamic designs: the first design was published in 1947 by Lane [15]; Reil and Hallett [16] later improved on it by using time-controlled air pulses instead of a continous flow.

3.1.1 Stry design

We performed a large number of initial tests using a device based on a design by Stry [2]. Fig. 3.1 shows a schematic drawing. The droplet generator is milled from a solid block of aluminum. A vertical through hole is drilled 20 mm from the left edge, and a second horizontal hole is drilled from the side to join it as shown. The latter is widened and tapped to accomodate a pressurized air supply hose. The vertical hole holds a blunt hypodermic needle, which is held by a plastic fitting screwed into a second aluminum block which is bolted onto of the first. Because the needles occasionally get clogged and need to be replaced, this setup is an improvement over Stry's monolithic design.

Our experiments showed that the ability of the instrument to produce droplets below 600 μm depends entirely on the precision with which the flow of water and air can be controlled. Regulating the fluctuating air pressure supplied by the building with but a simple valve, we were unable to reliably produce monodisperse droplets below 600 μm for more than a few seconds at a time.

While we decided not to pursue this approach further, it is worth noting that Coggins and Baker [17] designed a more elaborate apparatus with variable air and liquid flow and adjustable needle position, and succeeded in producing significantly smaller droplets on demand.

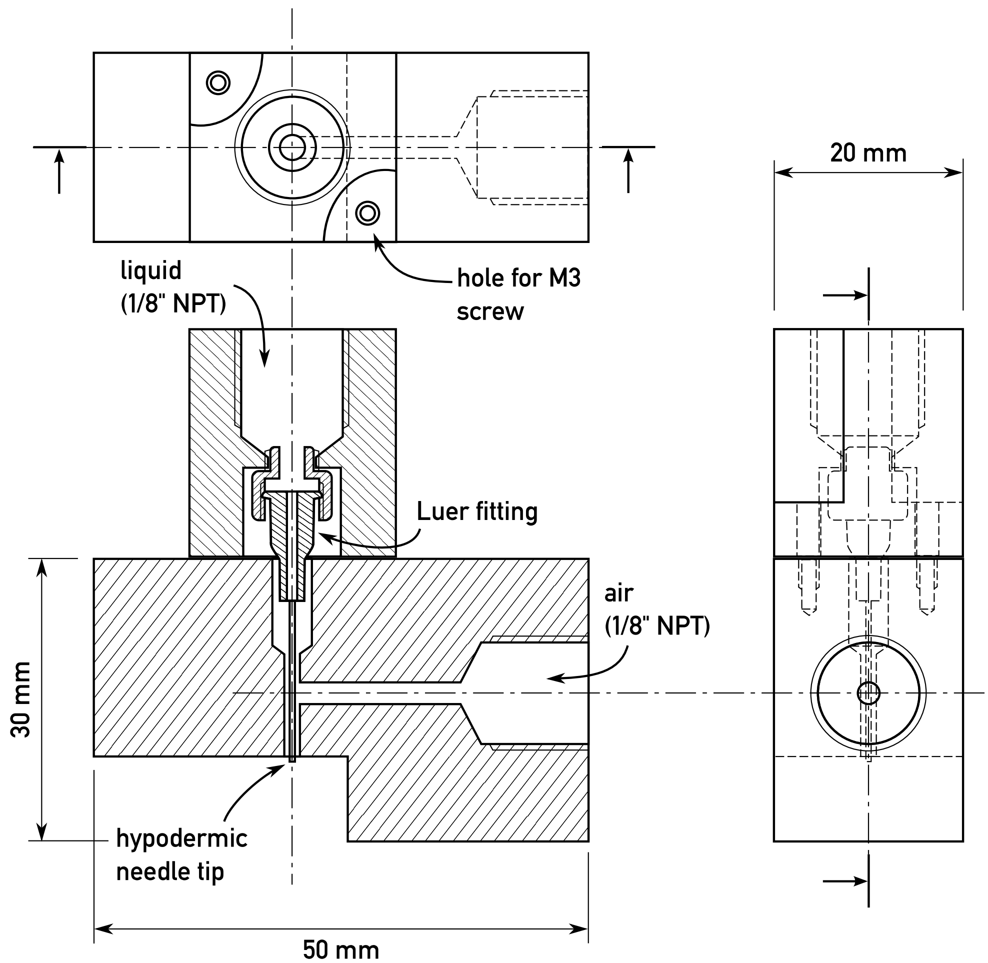


Figure 3.1: Schematic drawing of the coaxial-flow aerodynamic droplet generator, based on Stry [2]. Top: top view, right: rear view (third angle projection). Sectional view illustrates operating principle.

3.2 On-demand drop generators

Drop-on-demand technology finds its most important application in printing. Indeed, the most prominent designs representative of this category are the thermal droplet generators found in most household inkjet printers, invented by Endo et al. [7]. At least one research group, Sergeyev and Shaw [18], has succeeded in repurposing an old inkjet print head for laboratory droplet generation.

Less widespread, but more flexible in a research setting, are on-demand generators driven by the contraction of piezoelectric elements, such as those proposed by Yang et al. [19] or Ulmke et al. [20]. Excellent reviews on drop-on-demand designs were published by Le and Lee [21, 22].

An interesting third type of on-demand generator by Amirzadeh Goghari and Chandra [23] uses a short pulse of pressurized air, controlled by a solenoid valve, to eject a small amount of liquid through an orifice.

While drop-on-demand generators are a crucial component in applications like inkjet printing or microfluidics, they tend to suffer from aspired air bubbles, pileup of liquid around the nozzle tip, clogging, and other issues thwarting reliable drop expulsion unless manufactured and operated with great attention to detail.

3.2.1 Amirzadeh Goghari and Chandra design

We constructed a droplet generator based on the design by Amirzadeh Goghari and Chandra [23]. While both its construction from off-the-shelf parts and its operation are remarkably straightforward, it has two limitations:

- the duration of the air pulse is limited by the response time of the solenoid valve used. The shortest pulse we were able to reliable produce was on the order of a few milliseconds, which did not permit us to produce droplets smaller than a few hundred microns in diameter, and
- the head of water over the orifice must be kept very low to prevent leakage. As a result, the number of droplets that can be ejected is limited before the water needs to be replenished.

Owed to our lack of access to an automatic micropipette puller, the nozzles used in this experiment were not optimal, which likely contributed to our experience of frequent satellite droplets and liquid buildup at the nozzle tip.

3.2.2 Modified Yang design

A popular piezoelectric-based drop-on-demand design was proposed by Yang et al. [19]. It consists of a liquid-filled chamber, one wall of which is the underside of a piezoelectric disk—a brass disk coated with a circular piece of piezoelectric material, commonly found in electric buzzers.

To evaluate the performance of such a drop generator, we constructed several modifications of it, the final one of which is shown in Fig. 3.2. To make the chamber as flat as possible, minimizing the distance between piezoelectric disk and orifice, it has a depth of only about 2.5 mm, the thickness of a sheet of acrylic. A second sheet holds the disk in place, while a third sheet makes up the bottom wall of the chamber. Nozzle and inlet are glued directly into the bottom sheet.

To operate the droplet generator, water is fed through the inlet port until the chamber is filled and all air bubbles have escaped through the upward-facing nozzle. The generator is then turned so that the nozzle faces down and 30 ms pulses of about 30 V are delivered to the piezoelectric disk.

The greatest challenge faced was the accumulation of liquid on the nozzle surface, which quickly led to satellite droplets or thwarted droplet production altogether. This was a problem particularly with the

Again, capillaries drawn with an automatic pipette puller are likely more resistant to this effect.

3.2.3 Piezo-based drop generator

We tried building a Piezodropper from old piezoelectric elements squeezing glass capillaries (just like in the Ulmke paper), but two of the piezo elements were broken, and the third one had a capillary that clogged up repeatedly. We abandoned the approach before building a functioning droplet generator, although it seems attractive in practice. One downside is that the generation of amplified signals isn't straightforward—we used a soundcard connected to an amplifier to generate the signals.

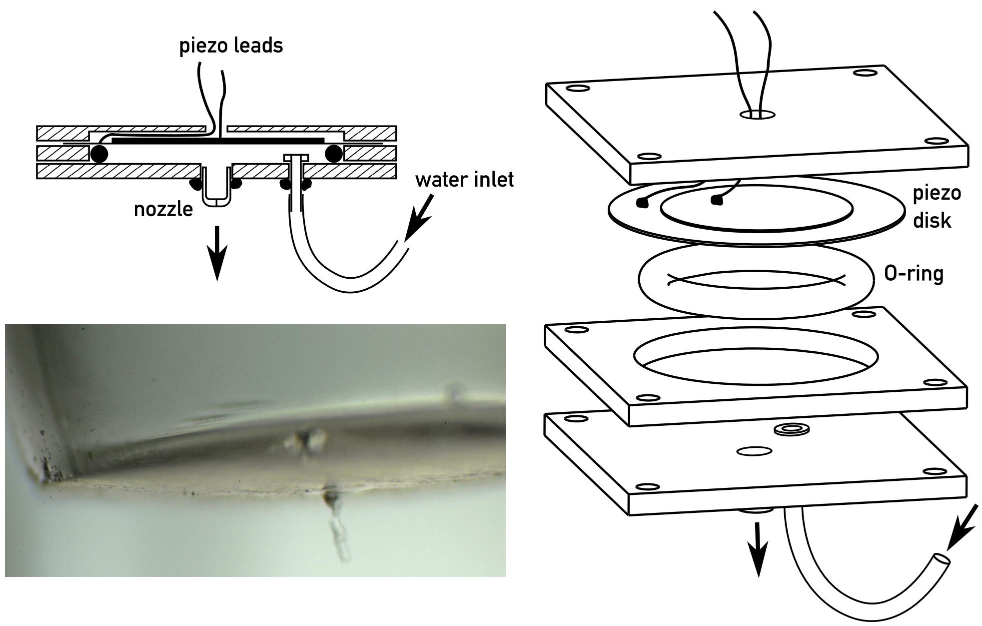


Figure 3.2: Top and right: schematic cross-section and exploded view of our piezoelectric droplet generator. Bottom left: photomicrograph of the nozzle tip ejecting a column of water (diameter $\approx 125\,\mu$m), which is about to coalesce into a round droplet.

3.3 Continuous-stream droplet generation using a computer hard drive

The content presented in this section has, in edited form, been accepted for publication in the journal "Review of Scientific Instruments" under the title "A simple vibrating orifice monodisperse droplet generator using a hard drive actuator arm".

A third category of droplet generators comprises devices operating on the basis of *Rayleigh breakup*, i.e. the disintegration of a disturbed round liquid jet into single droplets. The physics behind this phenomenon have been understood for almost two centuries—they were first studied by Savart [25] and later explained by Rayleigh [26]. Waves on the free surface of the jet can grow to a critical point at which those surface tension forces promoting pinch-off overcome those restoring the smooth jet. When the waves are induced by carefully controlled mechanical vibrations at an appropriate frequency, the droplets will be of uniform size and evenly spaced.

While it is possible to induce the disturbances using coaxial [24] and even perpendicular air flow,[1] it is far easier to create them by vibrating the orifice. This simple principle has been employed to generate droplets for fifty years, with orifices typically attached to either one of two vibrating mechanisms: an ordinary loudspeaker, first used by Donnelly and Glaberson [27], or a piezoelectric element, as first proposed by Schneider and Hendricks [28] and popularized by Berglund and Liu's design [29].

We first used a large plastic underwater woofer with a bendable metal strap taped to the cone. The end of said metal strap touched and vibrated the nozzle. This method is effective when the nozzle is relatively large (diameters $\geq$ 1 mm) and frequencies are in the 100s of Hz. Thinner jets require higher frequencies to break up evenly, but at tones above 1 kHz the amplitude needs to go up considerably to ensure a reliable breakup. Taking into account the response curve of human hearing, this means that the generation of sub-millimetre droplets jeopardizes the laboratory peace.

Most commercially sold droplet generators employ the principle of Rayleigh breakup by piezoelectric vibration. Berglund and Liu [29] proposed a design featuring a jewel orifice mounted into a machined cup sitting on a piezoelectric ring element. Because they

[1] The finding that a crossflow of air can, under the right conditions, induce regular Rayleigh breakup is a recent result of a separate research project on gas-water impingement that is yet unpublished.

lack a speaker's cone, piezoelectric elements are barely audible even when they oscillate at higher frequencies and amplitudes. Although these devices are well-suited to (and, in fact, often marketed towards) the calibration of sizing instruments—they typically include digital controls and a pressurized-air system to disperse the droplets into a cloud—they are too expensive for occasional laboratory use or early-stage prototyping projects.

As a result, we felt compelled to consider alternative sources of vibration that require a minimum effort to build and install using standard lab equipment.

3.3.1 Advantages of hard drive actuators

We propose that the actuator mechanism found in every magnetic hard drive is an optimal low-budget candidate for precision oscillation needs:

Very low cost. With high-capacity and solid-state devices rapidly pushing older hard drives into obsolescence, it should be a simple matter to acquire a few decommissioned specimens for demolition. Hard drives come in two form factors—3.5 and 2.5 inches wide, respectively—and both can be used for the purposes of this paper.

Further, glass needle orifices fabricated for use with existing loudspeaker setups can be reused, and are easily produced by hand from heated borosilicate capillaries or using a micropipette puller. The process is illustrated in Fig. 3.3 and in-depth instructions are given by Lee [22]. When needles with Luer fittings are used, the nozzles become exchangeable. Heated glass capillaries will have round orifice edges, which according to Dressler and Kraemer [30] leads to a noticeable variance in discharge coefficient and thus drop volume, but we have not been able to confirm this problem.

Piezoelectric-based devices, on the other hand, need fitted orifices to produce a range of drop sizes.

Ease of construction and installation. Unlike loudspeakers, hard drives have a flat base plate which can be drilled into, allowing for easy installation on any experiment jig. Save for a drill and a saw, no machining tools are needed for the construction of the droplet generator.

High amplitudes without noise. Like piezoelectric elements, vibrating actuator arms are very quiet, enabling use at frequencies and amplitudes that far exceed responsible levels on a speaker. In our experiments, the actuator responded to frequencies throughout our

Figure 3.3: Above: assembly of nozzle from low-gauge hypodermic syringe (Luer fitting) and capillary. Below: nozzle tip fabrication, capillary from left to right: broken, sanded, heated in a flame (I.D. 200 μm), heated for longer (I.D. 25 μm, could be sanded down by about 200 μm), overheated (I.D. 0 μm).

hearing range—i.e., up to 17 kHz—and likely well beyond, though we have not tested the full response range.

As an added advantage over other designs, no amplification is needed. Below 100 Hz, amplitudes on the order of 0.5 cm are easily achieved (albeit they are of course not needed for droplet production) when a peak-to-peak voltage of $2 - 4$ V is applied. The amplitude scales down with the inverse of the frequency, however, such that they are much smaller at typical operating frequencies ($0.5 - 10$ kHz). Nevertheless, the voltages required are well within the ability of any standard laboratory function generator; they can likely even be produced by many consumer-level computer sound cards.

3.3.2 Operating principle

Magnetic hard drives store data as sub-micron-sized patterns of oppositely magnetized dots on disks called *platters*. The read-write head is mounted at the tip of an arm that pivots across the platter surface while the platter spins. This setup allows the head to access the entire platter surface.

Fig. 3.4 illustrates schematically the design of a typical rotary actuator arm assembly. The flat voice coil mounted on the surface is responsible for the arm's side-to-side movement: as it is positioned under a permanent magnet, the coil creates a sideward force

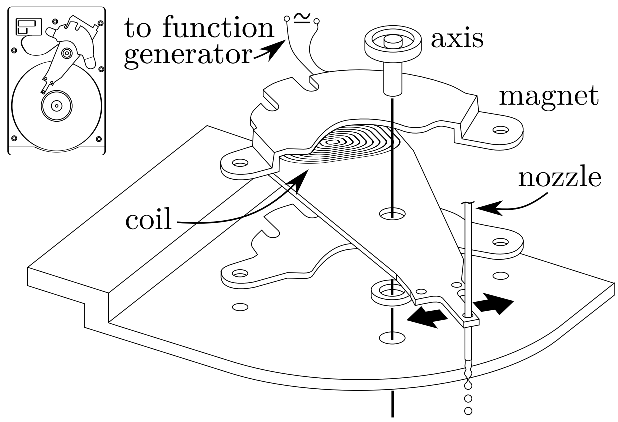


Figure 3.4: Top view of a hard drive and exploded view of the cut-out base plate, actuator arm, axis, and magnet assembly.

(*Lorenz* or *Laplace* force) when a current flows through its wires. The force is aligned with the cross product of the current direction and the magnetic field lines. By stopping or reversing the current, the arm's motion is likewise stopped or reversed. An applied sinusoid signal can thus be used to make the arm oscillate laterally at the desired frequency.

Since a typical hard drive's platter spins at up to 7200 RPM, actuator arms must be able to move with extreme speed and precision. They are thus engineered to be very light yet stiff. These characteristics make a magnetic hard drive's actuator arm an ideal supplier of in-plane vibrations. Indeed, hard drive actuators are remarkable not for their operating principle but for their low cost; it is only the economics of mass manufacturing that has in recent years enabled these high-speed, lightweight precision mechanisms to become so widely available.

3.3.3 Construction

If possible, forgo multi-platter drives, as they are more cumbersome to disassemble and have bulky, complex actuator assemblies. The device shown in Fig. 3.5 is based on a single-platter drive.

Dismantle and cut. After removing the hard drive cover, remove the top magnet, arm axis (1 in FIG. 3.6), arm (2), ribbon wires (3), circuit boards (4), and platters (5) such that only the base plate remains. Now the corner of the base plate holding the actuator arm assembly can be cut out (dashed line) to yield a shape as shown in FIG. 3.4. A band saw,

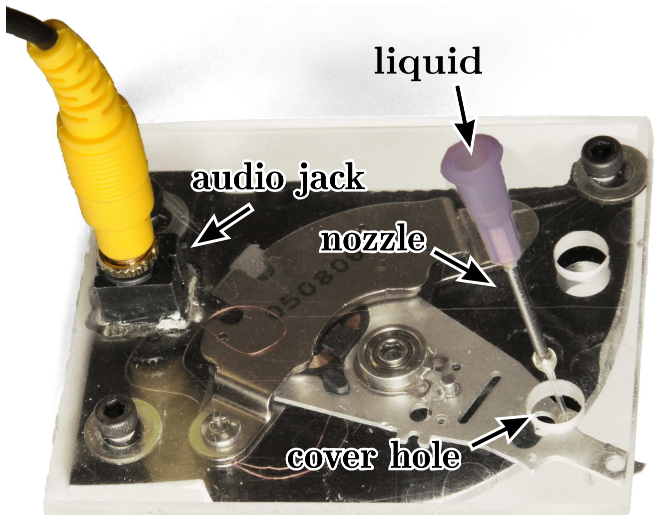


Figure 3.5: A fully assembled droplet generator from a single-platter drive. Nozzle is shown as inserted through one of the holes in the actuator arm.

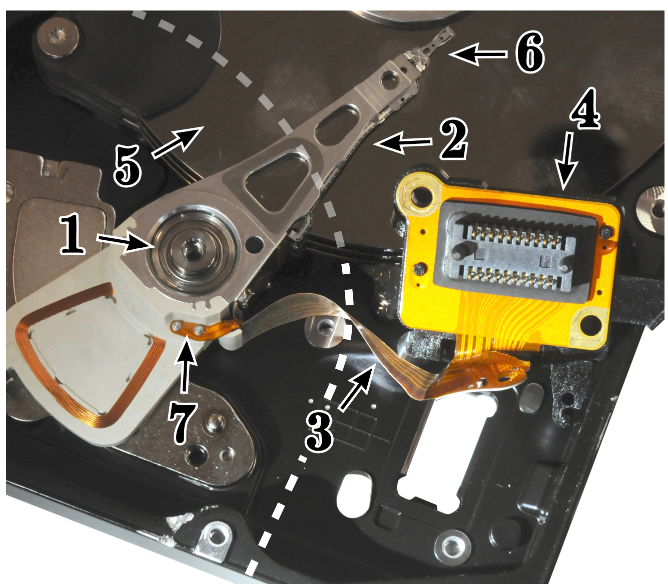


Figure 3.6: A multi-platter hard drive after removing cover and detaching circuit board (4). Top magnet is removed. Base plate can be cut along dashed line.

jigsaw or powered hacksaw will be very useful, although not necessary. The goal is to allow the tip of the arm to protrude over the edge. After cutting, reinstall the arm, axis, and top magnet.

Expose coil leads. Next, remove the read/write head (6) and all wiring leading to it, along with any connected I/O and servo circuitry (4). Be careful, however, not to tear off the two strands powering the voice coil, which are often integrated in the same ribbon cable (3). If you wish to remove the latter, ensure that exposed terminals (7) remain onto which you can solder new leads.

Add protective cover. We recommend bolting on a cover plate, such as a small sheet of transparent plastic, to protect the protruding arm from accidental bending. Drill a hole through the cover to allow the nozzle to be threaded through the arm. A severable connection from coil to function generator is preferable to a direct wire, if only because the voice coil leads are delicate and easily torn off. To this end, we epoxied an audio jack into the cover plate (see FIG. 3.5) and soldered the voice coil leads to it from the bottom.

3.4 Operation

To use the droplet generator, simply insert a nozzle through a small hole at the tip of the actuator arm—typically at least one hole will already be present where the read/write head was installed—and connect the voice coil leads to the output terminal of a function generator set to an initial peak-to-peak voltage of 1 V and a sinusoid frequency of about 50 Hz, which should cause weak but perceptible oscillations.

Interchangeable nozzles made from needles with Luer fittings, as in Fig. 3.3, are convenient and can be held in place by a male adapter clamped into a lab stand.

The nozzle must be supplied by an accurately calibrated syringe pump. It is convenient to integrate a large liquid reservoir (or tap water hose) via a T-valve between the between pump and nozzle to permit quick topping up of the syringe. In such a setup ensure that the reservoir valve is shut closed before operation, since pressure fluctuations at the nozzle are the most common culprit for unstable jet breakup conditions.

As with other vibrating orifice droplet generators, it is crucial that stable conditions are established before any experiments can begin. First, confirm that the liquid is ejected in a single jet. Multiple jets can be due to a clogged orifice (a mixture of distilled water and CLR®, drawn back through a syringe, is an excellent remedy). Satellite droplets can also form secondary jets, in which case the oscillation frequency must be adjusted or the amplitude reduced. Satellite formation is easily detected by using a gentle air flow to deflect the jet—if the droplets are truly monodisperse, they will all deflect at the same angle.[31]

3.5 Determining the produced droplet size

Note also that the orifice diameter D_o dictates the range of viable frequencies f as

$$3.5 \lesssim \frac{Q}{\pi f \left(\frac{D_o}{2}\right)^2} \lesssim 7, \tag{3.1}$$

where Q is the flow rate [25, 26]. In practice, D_o need not be precisely determined; it is easy to find appropriate settings for Q and f by viewing the jet against a strobe light, adjusting flow rate for a breakup length on the order of $10D_o$ (empirically for water), then tuning the frequency until droplets appear evenly spaced and spherical. It can be helpful to mount a magnifying lens in front of the orifice, as the adjustment procedure can become tedious when the droplets are very small.

Under stable conditions, every oscillation of the nozzle will produce one droplet downstream, such that the droplet diameter, assuming perfect sphericity, will be

$$D_d = \sqrt[3]{6Q/(\pi f)}. \tag{3.2}$$

This can be verified photographically, as shown in Fig. 3.7.

We have successfully used the generator to produce droplets of $0.1 - 1$ mm diameter, but both smaller and larger droplets are feasible.

3.5.1 Photographing droplets

With drop-on-demand approaches, the diameter of the produced droplet is more difficult to predict, particularly since not the whole squeezing volume may result in ejected liquid (e.g. with the Chandra generator, it's just a small droplet). So to find out just how large these droplets are (or just to verify the accuracy of Eq. (3.2)), we must resort to photographic means.

While there are different methods of photographing droplets, just taking pictures in

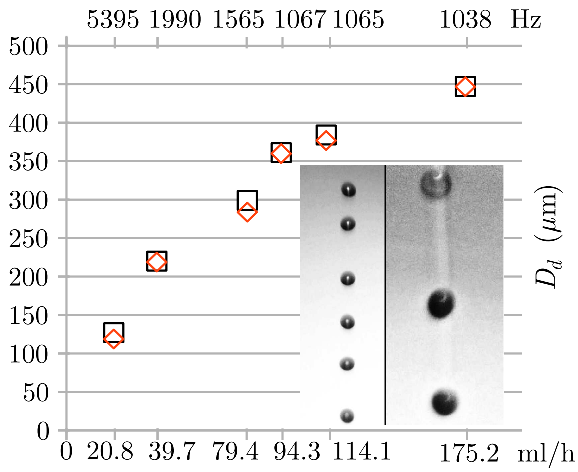


Figure 3.7: Some frequency/flow rate conditions under which stable droplets were produced (□ as predicted by Eq. 3.2, ◇ as estimated from photographs). Photos shown are $D_d = 200\,\mu$m and $D_d = 386\,\mu$m, respectively.

front of a strobe light has worked well. We used a millimetre scale (see Fig. 3.8) placed next to the droplet stream, and then used a conversion between pixels to find the size. Of course, this method suffers from barrel distortion that inevitably happens, especially with a zoom lens. The coefficients of this distortion are known and publically available for most lenses. We have not applied this correction in our photos, since the error introduced by blurred edges is likely more significant.

Flash lighting is best applied through a diffusing screen placed behind the droplets. Standard flash lamps may not always be fast enough—in that case, gas discharge strobe lamps serve the same purpose.

3.5.2 Droplet collisions

No droplet generation mechanism is perfect. Small fluctuations in flow rate, unwanted harmonic vibrations and air turbulence can cause disturbances in the stream of evenly spaced droplets—the smaller the droplets, the more often this happens. Occasionally, this will lead to the collision of two droplets some distance away from the orifice.

When two drops of diameter D_d collide, the diameter of the new droplet equals

$$D_{d+d} = 2\sqrt[3]{2\left(\frac{D_d}{2}\right)^3} = \sqrt[3]{2}D_d \approx 1.26D_d. \tag{3.3}$$

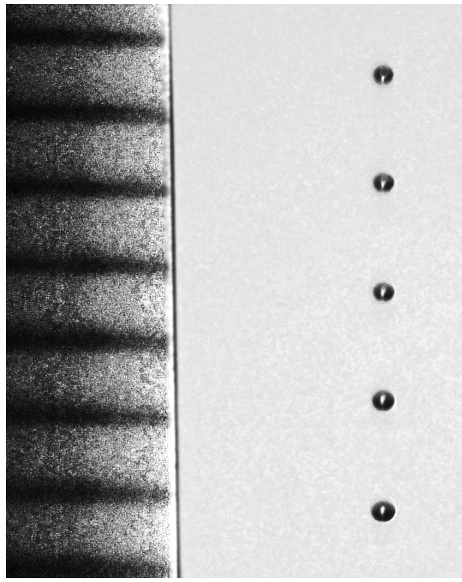


Figure 3.8: Scale besides droplet stream.

Indeed, secondary peaks will often appear in diameter histograms at precisely 126% of the peak diameter. As long as the underlying phenomenon is understood and kept under control, these secondary peaks should be no cause for concern during the calibration. Typically, photographs will confirm that a few droplets go astray and collide with others. Since the "real" diameter peaks are easily discerned, the secondary peaks can simply be ignored.

ILIDS

Interferometric Laser Imaging for Droplet Sizing (ILIDS), also known as IPI (Interferometric Particle Imaging) and MSI (Mie Scattering Imaging) is a popular optical droplet sizing method in which a spray is illuminated by a sheet of laser light and the scattered light is imaged laterally. The laser light is both reflected and refracted by the droplets, such that each droplet produces a pair of apparent "glare points". When seen through a lens away from the focal plane, each pair of glare points (the points being sources of coherent monochromatic light) appears as an interference pattern which, after falling through a circular aperture, casts an image that is a circular disk of fringes. The spatial frequency of the fringes is (to a very close approximation) linearly related to the droplet size. The phenomenon was first described by König et al. [32] and later in greater detail by Glover et al. [33]. Turnkey ILIDS setups for spray characteriziation are now widely available, comprising typically a pulsed Nd:YAG-laser, one or two CCD cameras, a timing circuit, and a piece of image processing software.

4.1 Operating principle

The number of fringes N_{fr} appearing in the image has a simple linear relationship to the droplet diameter D_d:

$$N_{\mathrm{fr}} = \varkappa D_d, \tag{4.1}$$

where $\varkappa$ is a constant derived from the optical configuration:

$$\varkappa = \frac{\arcsin\left(\frac{D_a}{2z}\right)}{\lambda}\left(\cos\frac{\varphi}{2} - \frac{m\sin\frac{\varphi}{2}}{\sqrt{m^2+1-2m\cos\frac{\varphi}{2}}}\right). \tag{4.2}$$

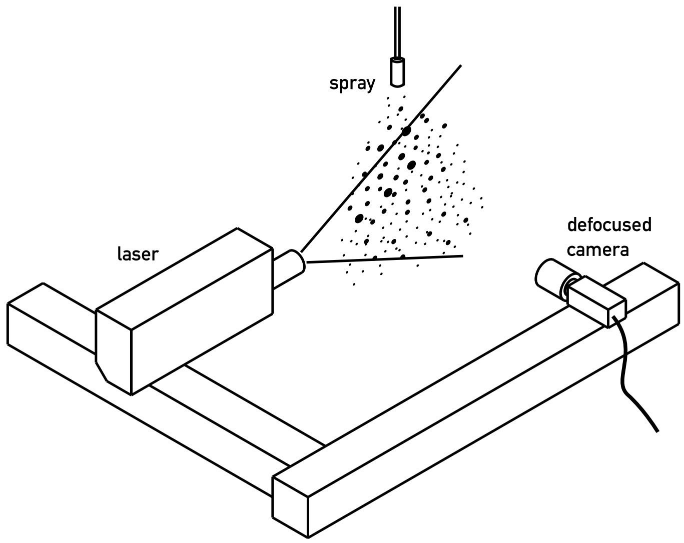


Figure 4.1: Perpendicular ($\varphi = 90°$), single-camera ILIDS setup

In the above expression D_a is the aperture diameter, z is the distance of the lens to the laser sheet, φ is the off-axis angle (90 degrees in most setups, including ours), and m is the relative refractive index of the droplets (1.333 for water in air).

As a consequence of geometrical optics, the distance s_x (in pixels) between two adjacent fringes has a linear relationship with the defocussing distance Δz, where M is the magnification, $d_{p,x}$ is the physical size of a camera sensor pixel, and $\Delta\vartheta$ is the angle subtended by two adjacent fringes entering the lens [34]:

$$s_x = \frac{\Delta\vartheta\Delta z}{Md_{p,x}} \tag{4.3}$$

Of course, equation (4.3) is only meaningful where $\Delta z \gg 0$. When the image is brought into focus ($z \approx 0$), fringes will give way to a sharp image of the glare points.[1] Of course, if the pixel density is too low to resolve both glare points, a single bright spot will appear.

[1]To be exact, diffraction will cause every point to be imaged as an Airy disk, but we shall neglect this effect here.

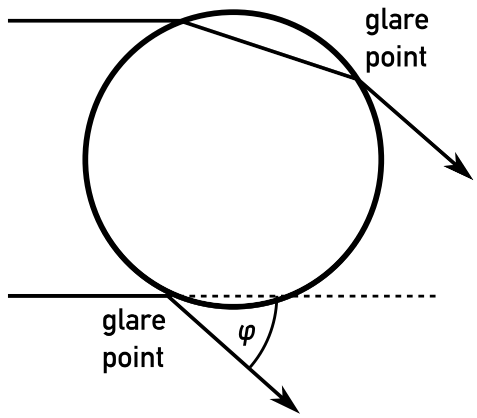


Figure 4.2: Reflected and first-order-refracted light rays, producing two glare points when viewed from an angle φ.

4.1.1 Influence of the scattering angle φ

The scattering angle φ, illustrated in Fig. 4.2, determines the relative contribution of different scattering orders of light to the imaged fringe pattern. Both geometric optics [9] and Mie theory provide methods to compute the total scattered intensity for a given φ and m; some examples can be found in Kawaguchi et al. [35] and Mounaïm-Rousselle and Pajot [36]. The geometric analysis approach is not valid beyond $\varphi > 70°$, as the first-order scattered beam ($p = 1$) is not visible from this angle [33].

While authors have identified several forward angles as optimal for their applications, e.g. $\varphi = 45°$ [33] or $\varphi = 66°$ [36], such configurations inevitably result in a variation in z, and therefore in a variation in the degree of defocusing, across the image unless the camera itself is similarly angled with respect to the lens to correct for this aberration (the so-called *Scheimpflug condition*). Since the latter approach requires specialized optical equipment, $\varphi = 90°$ is used in many setups, including the one in this paper, although it is far from ideal in terms of scattered light intensity (culminating in a suboptimal signal-to-noise ratio).

4.1.2 Optical limits on fringe detection

Optics impose theoretical and practical size limits on the droplets to be measured. We will outline them in the following paragraphs; the reader is referred to Damaschke et al. [37]

for a more detailed analysis.

Nyquist criterion for the fringe density. The Nyquist criterion requires that for the camera to be able to resolve a pair of neighbouring fringes, their images must be at least two pixels apart. This can easily be achieved by a sufficient defocusing the lens, which widens the fringe image, increasing the number of pixels covered by each fringe. The lens mechanics permitting, any arbitrarily large droplet can thus be measured after a quick adjustment. In theory, this correction is effective until the defocused droplet image is too large for the CCD sensor, and fringes are cut off. In practice, overlap and noise (see below) will cause significant problems long before the image can be defocused beyond the sensor edges.

Signal-to-noise ratio. Image noise is a significant source of trouble in ILIDS analysis. Indeed, many droplet images must be discarded as data sources because they are too weak compared to the noise. Small droplets suffer from this more than larger ones because they scatter less light,[2] but the problem also occurs with deeply out-of-focus images of very large droplets, as dilated droplet images spread the same amount of light over a greater area on the camera sensor. As a result, they are darker on average than less defocused images.

Minimum droplet size. Damaschke et al. [37] argue that the smallest measurable droplet is one that produces exactly one fringe falling through the aperture. We may speculate that, at least in theory, the fringe frequency should be measurable even if only a partial fringe is shown. This would require its image to be sufficiently zero-padded before the Fourier transform is applied to it. In practice, however, the intensity of scattered light typically drops below an acceptable level well before the fringes become too large, and noise (see above) will become the overwhelming problem.

Discrepancy between geometrical optics and Mie scattering for small droplet sizes. ILIDS users should also be aware that the assumptions of geometric optics that underlie (4.2) do not hold for small droplets. Mounaïm-Rousselle and Pajot [36] found that for isooctane droplets ($m = 1.39$) below $10\,\mu$m, geometric optics yield a fringe spacing value about 14% higher than that predicted by exact Mie scattering simulations at $\lambda = 532$ nm. While the deviation quickly vanishes for larger droplets, it is nevertheless noteworthy in the context of potential error sources.

[2]The scattered intensity grows with the cross-section of the droplet

Overlapping droplet images. The ability to image a whole 2-D field of droplets all at once is ILIDS' strongest selling point, yet also its curse. When droplets are spaced too closely and the lens is sufficiently defocused, the defocused disk images overlap and it becomes difficult to determine the fringe counts corresponding to individual droplets. Damaschke et al. [37] provide a statistical estimate on the fraction of overlapping disks (overlap coefficient).

4.2 Types of ILIDS setups

We shall now discuss, besides the standard ILIDS technique, two common variations: optical compression, which alleviates the problem of disk overlap, and double-camera setups, which allow for the simultaneous capture of ILIDS and PIV imagery, to be used to provide velocity information and/or droplet centre positions that may be used to identify overlapping disk regions.

4.2.1 Standard ILIDS

The most simple ILIDS configuration, as shown in Fig. 4.1, consists of a single digital camera with a defocused objective lens, placed at a right angle to the laser sheet. The lens aperture is (approximately) circular and typically completely open to permit as much light as possible to fall on the sensor area.

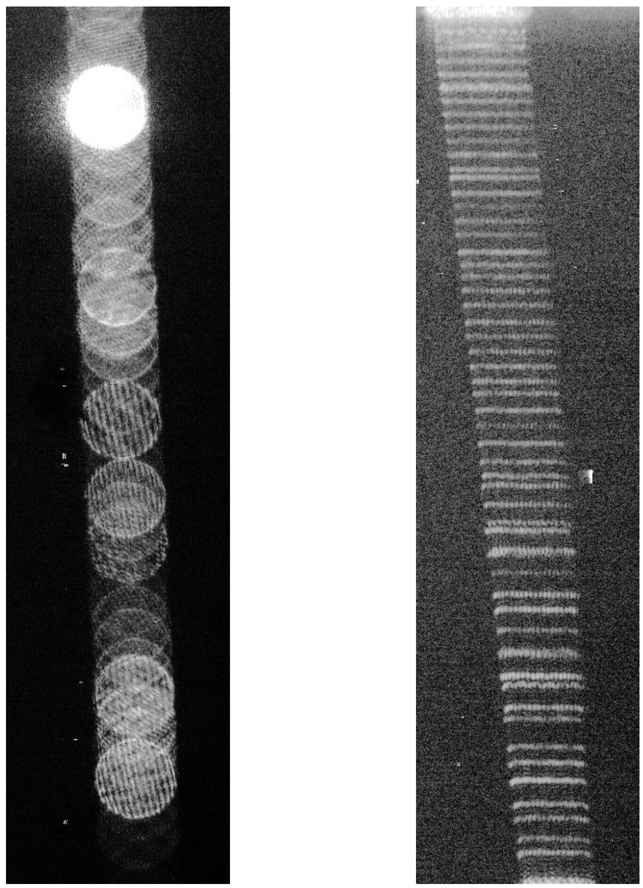
Both camera and laser are connected to a computer via a timing circuit, and both can be triggered simultaneously by software installed on the computer. Commercial ILIDS vendors provide the timing circuitry and the software, which typically integrates a collection of image processing algorithms that can be used to analyze the captured images immediately.

A series of images is taken in rapid succession, and the software uses a circle detection algorithm (correlation, circular Hough transform, or other) to identify disks that fall within a certain radius range. The dominant frequency is found for each disk and converted to a drop size.

The conversion factor, as evident in (4.2), depends largely on the collection angle of the lens, which is a function of z and D_a. Since it can be infeasible to measure these values directly, it is a good idea to derive these values empirically, as in Section 4.3.1.

The ability to image a whole 2-D field of droplets all at once is ILIDS' strongest selling point, yet also its curse. When droplets are spaced too closely, their defocused disk images overlap and it becomes difficult to determine the fringe counts corresponding to individual droplets. Damaschke et al. [37] provide a statistical estimate on the fraction of overlapping disks (overlap coefficient).

4.2.2 ILIDS with optical compression



(a) (b)

Figure 4.3: Before (a) and after (b) installing the slit aperture. The aperture stop pares off the top and bottom halves of the defocused circles, leaving only a narrow center string in the middle.

The problem of disk overlap is never more apparent than in efforts to calibrate the system using a vibrating orifice droplet generator, as the droplets produced thereby are spaced very closely and produce heavily overlapping defocused images. Fortunately, there exists a simple and reliable technique to deal with this problem: a slit aperture, installed directly in front of the lens, masks the defocused droplet images such that only a thin strip across their center passes through the lens. The effect is shown in Fig. 4.3.

Arguably the most popular way to reduce the amount of overlap is the use of optical compression techniques, whether by means of a slit aperture [34] or a cylindrical lens [35, 38]. However, some techniques (e.g. Global Phase-Doppler [39] and intensity-analyzing methods [40]) or use cases (e.g. very low signal-to-noise ratios) require the full disk image to be available. In these cases, the standard approach is to identify the location and outline of each disk image, such that the fringe analysis can either be limited to non-overlapping regions or be otherwise modified to take overlapping fringes into account.

Naturally, equations (4.2) and (4.3) still hold.

Extracting the fringe counts from such an image is straightforward. First, we correlate the image with that of a single, solid bright rectangle which shares the approximate dimensions of a typical strip in the image. This operation yields intensity peaks centered over our regions of interest. We remove closely adjacent peaks, as they may represent questionable or overlapping strips. Compared to the sheer number of correctly identified strips, the number of legitimate data points lost this way is negligible. Fig. 4.4 shows the result of such an attempt at identifying the strips.

To find the number of fringes within the strip, we cannot rely on counting the number of dark/bright variations directly, as some of them may be lost in the noise. The spatial frequency of the peaks, however, taken together with the known and constant horizontal width of the strips, will produce a reliable fringe count. In the next step, our algorithm therefore applies the Fourier transform to each region of interest. To improve the accuracy of the method, three steps are performed before the Fourier transform is taken:

1. a weak (3×3) Gaussian blur is applied to the region (optional);
2. a Hanning window is applied to the region—both horizontally and vertically. This reduces the "sinc ringing" effect encountered when taking the Fourier transform of finite signals;
3. the region is padded with zeros in all directions to yield a larger input to the Fourier transform. In our application, the windowed and padded strip images had dimensions of 1024×1024 pixels. Zero-padding increases the granularity of the frequency spectrum, which can help with the correct identification of the peak frequency.

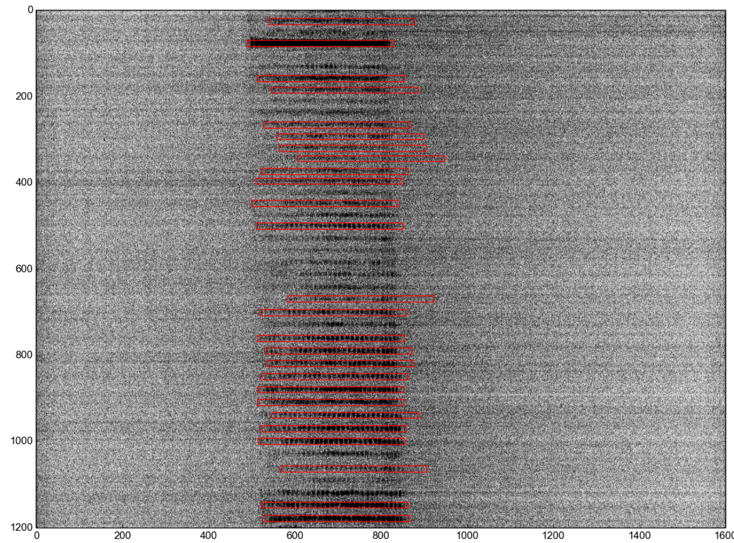


Figure 4.4: The image is correlated with that of a solid bright rectangle, which results in peaks that approximately coincide with the centers of the strips. Here, the original photo is shown with rectangles drawn centered at said peaks.

Fig. 4.5 shows the windowed appearance of one such region of interest (although it does not show the padded input to the Fourier transform due to space constraints). The Fourier transform yields a frequency power spectrum in two dimensions, although we are primarily interested in the frequency peak in the horizontal direction (i.e. along $y = 0$). In order to minimize the misidentification of dominant frequencies,

1. we clip the spectrum to a band of reasonable frequencies. This is necessary because a) $1/f$-noise causes very low frequencies to dominate in power, although they are of no interest to us, and b) graininess in the original photo can sometimes result in meritless high-frequency peaks;

2. we apply a Gaussian blur to the 2-D spectrum to remove outliers in the spectrum;

3. we discard all regions in which the peak frequency's power does not exceed a certain value;

4. we discard all regions in which the *prominence* of the peak freqency's power (i.e. its proportion to the mean power) does not exceed a certain value (this step is optional).

The bottom two elements in Fig. 4.5 illustrate the effect of these steps.

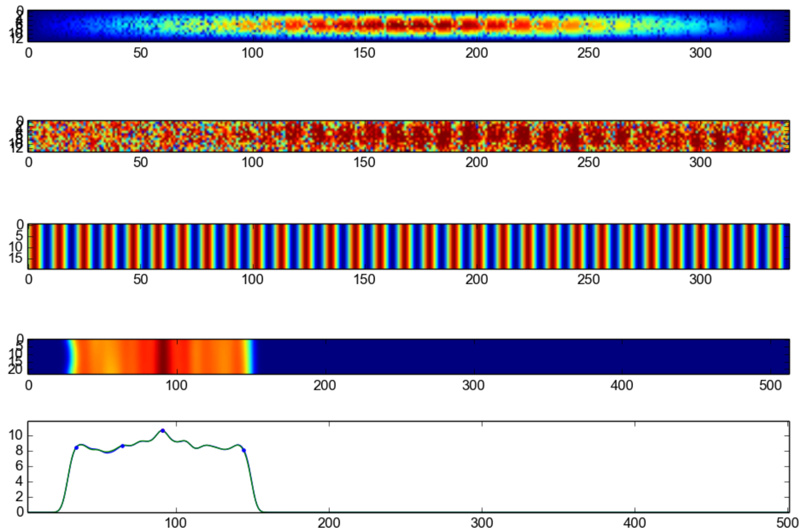


Figure 4.5: From top to bottom: windowed region of interest; original (unwindowed) region of interest; sine wave representing the identified peak frequency; clipped and lowpass-filtered 2-D frequency spectrum showing a distinct peak at about 90 oscillations across the image width of 1024 pixels; 1-D plot of the frequency spectrum, with peak identified at $f = 91.0$.

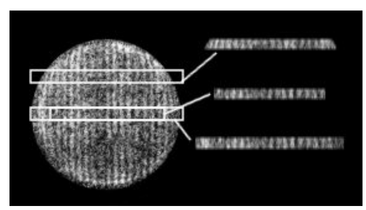


Figure 4.6: Only a slit aperture centered on the lens and extending across the entire lens entrance will preserve all fringes

Finally, the peak frequency f_{peak} is converted into a fringe count by re-scaling it from the padded size D_{padded} ($= 1024$ pixels) to the width of the strip (which, in the context of IPI measurements, should equal the diameter D_{i} of the defocused droplet image):

$$N_{\text{fr}} = f_{\text{peak}} \frac{D_{\text{i}}}{D_{\text{padded}}} \tag{4.4}$$

While the above algorithm will generally give a good estimate of the fringe count for a given defocused droplet image, it cannot know whether the entire center portion of the image has indeed passed the slit aperture. It is conceivable, after all, that the slit aperture was not perfectly centered on the lens entrance, or that the slit aperture was shorter than the diameter of the lens entrance. Fig. 4.6 illustrates how the slit aperture can cause the defocused image to appear smaller than it is. The reduced value for D_{i}, manually entered in equation (4.4), will result in droplets being reported as smaller

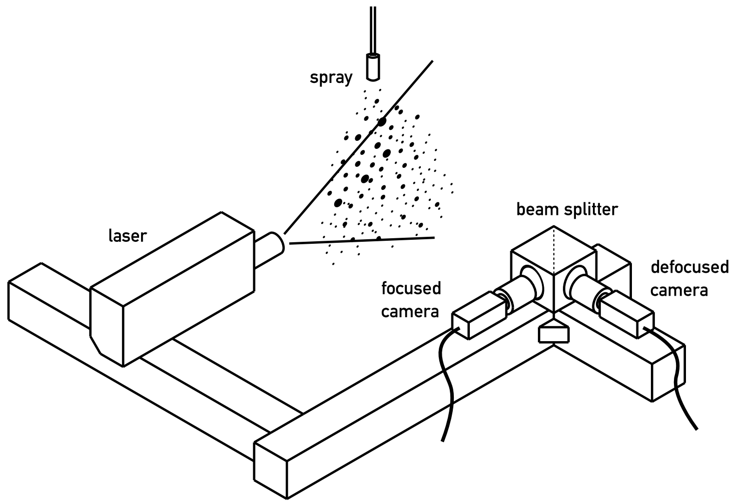


Figure 4.7: Perpendicular ($\varphi = 90°$), double-camera ILIDS setup

than they are in reality.

As noted above, ILIDS with optical compression can be used with an additional, focused camera behind a beam splitter, as shown in Fig. 4.7.[3] The latter may provide e.g. PIV images for a velocity analysis or LIF images for evaporation studies. This was demonstrated by Hardalupas et al. [41] and Hardalupas et al. [42]. However, such setups suffer from *center discrepancies*. Section 4.5 documents how such center discrepancies can be dealt with.

4.2.3 Additional focused image for disk detection

As described above, running a simple frequency analysis on bright regions in an ILIDS image is futile when two of the fringe disks overlap, as it is not clear how many droplets are associated with the region and where their disks infringe. While optical compression provides a very good solution to this problem, it is not always applicable: some techniques (e.g. Global Phase-Doppler [39] and intensity-analyzing methods [40]) or use cases (e.g. very low signal-to-noise ratios) require the full disk image to be available. In these cases, the standard approach is to identify the location and outline of each disk image, such that the fringe analysis can either be limited to non-overlapping regions or be otherwise mod-

[3]Alternatively, the two cameras may image the spray from different angles, e.g. as demonstrated by Glover et al. [33], although this complicates the setup as the Scheimpflug condition must be fulfilled.

ified to take overlapping fringes into account. Additionally, once disk centers are found, the software can apply Hanning windows to the disks as well as ignore disk pairs that are too close.[4]

To identify the location of overlapping disks in the image, an additional camera is introduced to capture a focused image of the spray simultaneously with the defocused ILIDS camera. The setup is identical to that shown in Fig. 4.7. The intensity peaks in the focused image are then taken to be the droplet positions (i.e. disk centers) in the defocused image.

The problem with this, as with all multi-camera setups, are center discrepancies. We show how to overcome that in Section 4.4.

4.3 Calibration of a system with optical compression

The conversion factor between fringe density and droplet size depends largely on the collection angle. Often, the relevant measure is the numerical aperture (NA), corresponding to the sine of half of the collection angle. For a simple lens,

$$\mathrm{NA} = \sin \frac{d_a}{2z}. \tag{4.5}$$

Most ILIDS literature, as well as the Dantec software, approximate the zoom lenses used in practice as a simple lens. Under this assumption, z is the distance from lens to light sheet, and D_a, the entrance pupil diameter, is a lens constant computed as (minimum focal length)/(minimum f-number). Modern zoom objectives, however, are not simple lenses: in many cases, both the physical (i.e. as measured inside the lens) and virtual (i.e. as seen from the front of the lens) diameter of the aperture change when the lens is focused. This change does not necessarily perfectly counteract the changing focal length to maintain a constant f-number throughout the zoom range. In other words, it is often impossible to determine the exact entrance pupil diameter of an arbitrary objective when it is focused onto an image at short distance.

By the same token, it is practically impossible to determine the true distance from the light sheet to the principal plane of the lens, since it is found somewhere *inside* of the lens

[4] The latter is an effective control mechanism, but is liable to skew the representativeness of the sample because small, dispersed satellites outside of the main flow are most likely to be validated.

body (altough generally, errors in z are less significant in relative terms than errors in D_a).

Taking these unknowns into account, it is advisable to run a few calibration tests with droplets of different sizes before employing the IPI technique for real spray measurements. Recall that, if we ignore the Mie error (Section 4.1.2), the relationship between fringe count and droplet diameter is linear with a constant of proportionality $\varkappa$ (see equation (4.2)). The aim of our calibration, then, is to determine the value of $\varkappa$ from experiment—the premise being that we cannot be certain of the values of D_a, z, and possibly not even m and φ (although the latter can usually be ascertained to a sufficient degree of accuracy).

4.3.1 A sample calibration of the slit aperture method

Using the droplet generator described in Section 3 and the IPI configuration described in Section 4.2, we produced and measured monodisperse droplets of many different diameters. The droplet diameters were ascertained both photographically and algebraically (i.e. from flow rates and vibration frequencies; see (3.2)), as described in Section 3.5. Out of over 30 sets of IPI measurements we selected for this exemplary calibration six sets that exhibited both strong uniformity and high photographic quality:

Set	Flow rate	Frequency	D_d, predicted	D_d, from photo	$\hat{N}_{\mathrm{fr}}$
FA	20.8 ml/h	5395 Hz	127 μm	126 μm	9.71
FB	39.7 ml/h	1990 Hz	220 μm	226 μm	16.71
FC	79.4 ml/h	1565 Hz	299 μm	291 μm	22.92
FD	94.3 ml/h	1067 Hz	361 μm	367 μm	27.26
FE	114.1 ml/h	1065 Hz	384 μm	384 μm	29.89
FF	175.2 ml/h	1038 Hz	447 μm	454 μm	34.56

Table 4.1: Six sets of calibration data taken with the setup described in Section 4.2

The values for $\hat{N}_{\mathrm{fr}}$, the peak fringe count, are based on the histograms (see Fig. 4.8) showing the distribution of fringe counts within each dataset. These fringe counts are of course found by the algorithm described in Section 4.2.2. Every image contained between 60 and 80 droplets.

It is worthwhile to point out some apparent idiosyncrasies in the histograms of datasets FB and FC. Their peak fringe counts are 16.71 and 22.92, but there are secondary peaks at about 21 and 29 fringes, respectively. The latter are explained by the collision of droplets as discussed in Section 3.5.2, and they are ignored for the purposes of calibration.

The close agreement of the droplet diameters found from photographs with those pre-

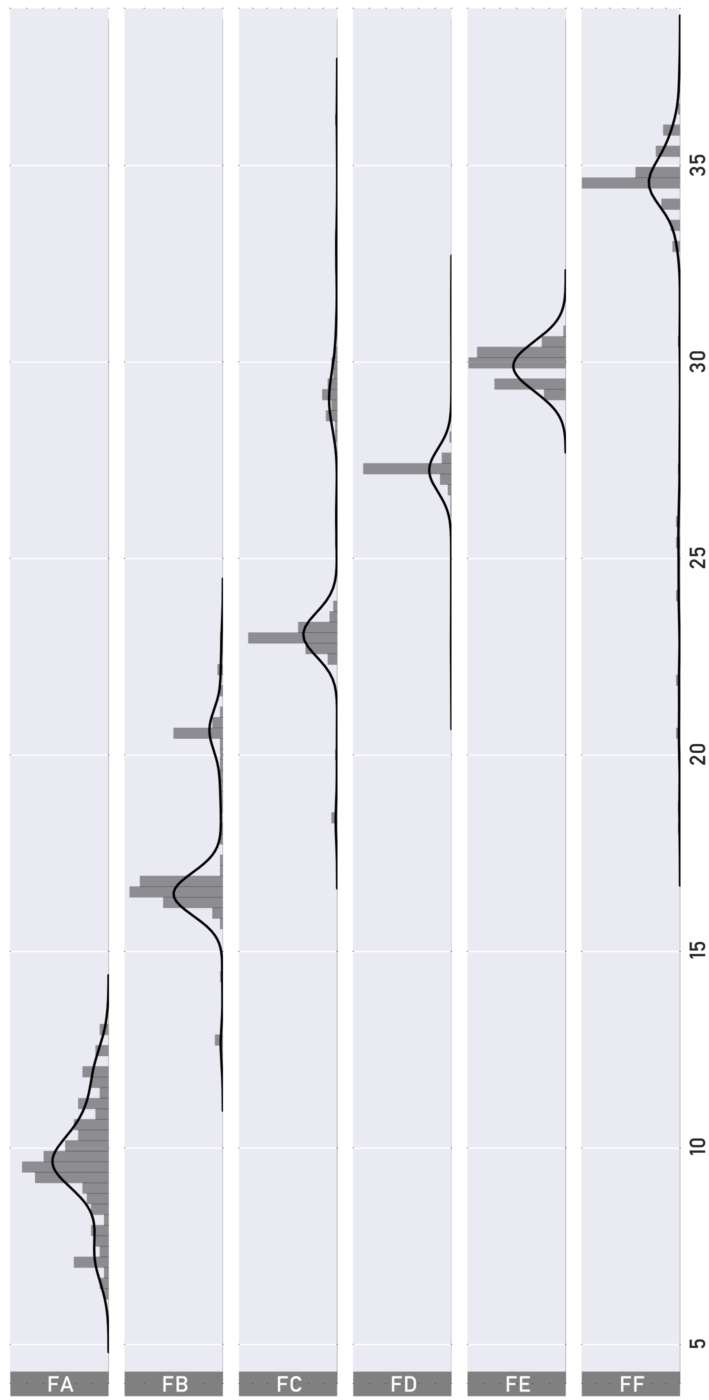


Figure 4.8: Normalized distributions of measured fringe counts N_{fr} for the six datasets listed in Table 4.1. Solid lines are Gaussian kernel density estimates with $h = 0.5$.

dicted by (3.2) reassures us that we can use the predicted D_d for further analysis.

At this point, we can least-squares-fit the linear relationship (4.2) to the primary peaks $\hat{N}_{\text{fr}}$ and the known droplet diameters D_d to find $\hat{\varkappa}$:

$$\hat{\varkappa} = \frac{\sum_i D_{d,i}\,\hat{N}_{\text{fr},i}}{\sum_i D_{d,i}^2} \tag{4.6}$$

Note that instead of the standard least squares regression we here use a simplified formula to force the trend line through the origin. This choice should not be made lightly, since it will usually cause the residuals to have a non-zero mean. In this case, however, we believe it to be justified to require that $D_d = 0$ for $N_{\text{fr}} = 0$.

Based on the values in Table 4.1, we thus arrive at a value of $\hat{\varkappa} = 76808.1$ with an R^2-value of 99.98%.

Fig. 4.9 illustrates the good agreement on $\varkappa$ between all datasets. Considering the sheer number of error sources—from the unavoidable non-uniformity of the generated droplets to the uncertainty that comes with taking the Fourier transform of a noisy image—the calibration results documented here are a testament to the practical robustness of the method.

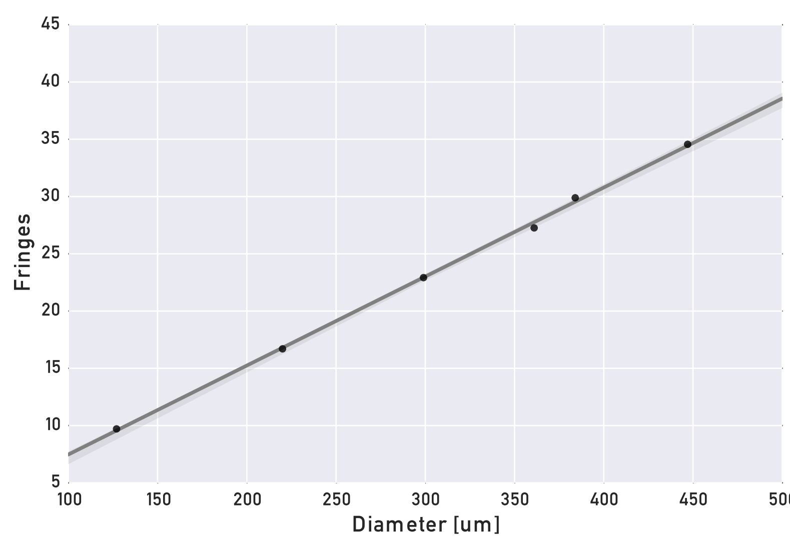


Figure 4.9: Scatterplot of Table 4.1 showing the peak fringe counts $\hat{N}_{\text{fr}}$ for each predicted droplet diameter D_d

It must be remembered, of course, that the peak fringe count values $\hat{N}_{\text{fr}}$ forming the basis of our calculation are taken from the peaks of Gaussians fitted to the raw fringe count histograms (see Fig. 4.8). In other words, it is our assumption that all droplets from a given dataset produce fringe counts that are normally distributed around their respective $\hat{N}_{\mathrm{fr}}$. The histogram to dataset FA shows a much higher deviation than the others—this may be due to genuine variance in the generated droplet diameters or to difficulty in processing comparatively weak images with low fringe counts. It seems likely that both effects contribute.

We can compare the empirically determined value $\hat{\varkappa}$ with the mathematical result obtained from (4.2). Substituting $\lambda = 532$ nm, $m = 1.3324$ and $\varphi = 90°$, we conclude that

$$\frac{D_a}{z} = 2\sin\left(\frac{\hat{\varkappa}\lambda}{\cos\frac{\varphi}{2} - \frac{m\sin\frac{\varphi}{2}}{\sqrt{m^2+1-2m\cos\frac{\varphi}{2}}}}\right) = 2\sin(3.11982\cdot 10^{-7}\hat{\varkappa}), \tag{4.7}$$

so our value of $\hat{\varkappa} = 76808.1$ yields $\frac{D_a}{z} = 0.047921$. Recall that this quotient is a measure of the collection angle and closely related to the numerical aperture NA= $\sin\frac{D_a}{2z}$. If needed, we can now use this result to compute the input parameters D_a and z in the DantecStudio IPI software: given, for instance, $z = 45.0$ cm, we can obtain the entrace pupil diameter as

$$0.047921 \cdot 450\,\mathrm{mm} = 2.156\,\mathrm{mm} \tag{4.8}$$

The above procedure will result in more accurate measurements compared to the simple measure-and-enter technique suggested in the manual accompanying the Dantec DynamicStudio software. It will need to be repeated whenever the lens is changed, re-focused, or moved with respect to the light sheet.

4.4 Removing centre discrepancies using keypoint registration

The content presented in this section is, in edited form, under review for publication in the journal "Measurement" under the title "Eliminating center discrepancies between simultaneously captured ILIDS and PIV images by means of direct homography estimation".

As mentioned above, to allow both cameras to image the same physical region in the spray, they are either placed behind a beamsplitter at a right angle to the light sheet, or

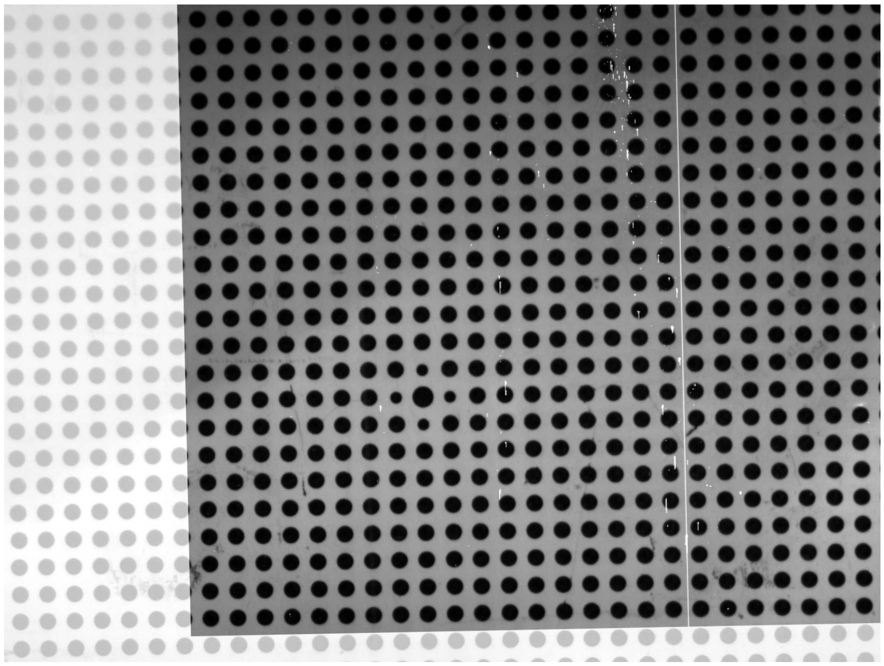


Figure 4.10: Homography **H** applied to target pattern image captured by the focused camera and superimposed on the image captured by the defocused camera (here, both cameras were in focus for the calibration only).

placed separately at different angles. The latter approach makes for a more difficult setup, since Scheimpflug's rule demands that the camera must be tilted with respect to the objective lens, although it gives the user the freedom to choose the highest-intensity scattering angle.

In any of the above cases, the use of two cameras requires that their images be mapped onto one another. This is commonly achieved by means of a camera calibration procedure, in which a target pattern (e.g. as in Fig. 4.10) of known dimensions is photographed by each camera. A pattern recognition algorithm then determines the object-to-image mappings for each camera:

$$\begin{bmatrix} x' \\ y' \\ z' \\ r' \end{bmatrix} = \begin{bmatrix} S_x & A_{yx} & A_{zx} & T_x \\ A_{xy} & S_y & A_{zy} & T_y \\ A_{xz} & A_{xy} & S_z & T_z \\ P_x & P_y & P_z & S_0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{4.9}$$

In our case, Dantec supplied a *standard dot target*, a white $10 \times 10\,\mathrm{cm}^2$ plate engraved with a pattern of black dots. The plate is to be mounted such that its surface coincides perfectly with the laser sheet.

In practice, $P_{x,y,z} = 0$ and $S_0 = 1$ is assumed, such that the mapping is affine. The z-components (third row/column) are further assumed to be zero, such that a 3×3 matrix suffices for the purposes of this discussion:

$$\begin{bmatrix} x' \\ y' \\ r' \end{bmatrix} = \begin{bmatrix} S_x & A_{yx} & T_x \\ A_{xy} & S_y & T_y \\ P_x & P_y & S_0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{4.10}$$

The calibration algorithm thus finds the camera matrices $\mathbf{P}_{\mathrm{foc}}$ and $\mathbf{P}_{\mathrm{def}}$ mapping the object coordinates $\mathbf{x}$ onto the two camera images $\mathbf{x}'_{\mathrm{foc}}$ and $\mathbf{x}'_{\mathrm{def}}$ (the respective subscripts shall hence designate the focused and defocused cameras):

$$\mathbf{x}'_{\mathrm{foc}} = \mathbf{P}_{\mathrm{foc}}\,\mathbf{x} \tag{4.11}$$

$$\mathbf{x}'_{\mathrm{def}} = \mathbf{P}_{\mathrm{def}}\,\mathbf{x}. \tag{4.12}$$

It follows that the quotient of the two matrices, also known as the homography

$$\mathbf{H} = \mathbf{P}_{\mathrm{def}}\,\mathbf{P}_{\mathrm{foc}}^{-1} \tag{4.13}$$

can be used to map the focused image onto the defocused image, as shown in Fig. 4.10:

$$\mathbf{H}\,\mathbf{x}'_{\mathrm{foc}} = \mathbf{x}'_{\mathrm{def}}. \tag{4.14}$$

Unfortunately, the calibration procedure itself introduces an unwanted distortion: to capture a viable photo of the target pattern, the defocused camera must be temporarily brought into focus, as was done in Fig. 4.10. This is not mentioned e.g. in the application manual of Dantec's IPI system, but is a practical necessity. Bringing a camera out of focus not only introduces a blur, it also scales the image extents. Fig. 4.11, adapted from Hardalupas et al. [41], shows schematically how this effect creates "center discrepancies". Since the extents of the defocused image are either smaller or larger than those of the focused image, depending on the direction of defocusing, all droplet images are projected either closer to or farther away from the image center. The discrepancy is worst for droplets far away from

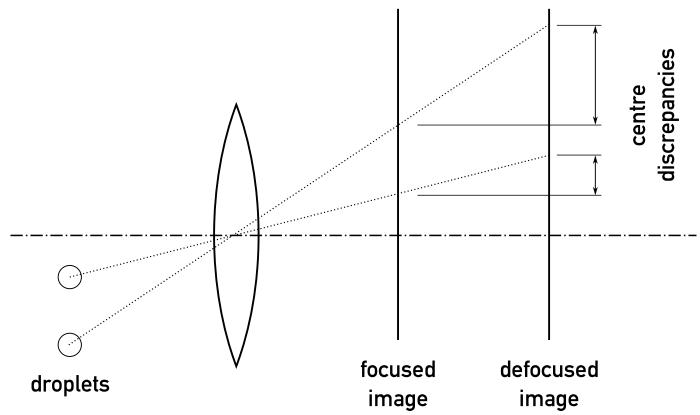


Figure 4.11: Schematic showing the source of center discrepancies in the case of parallel image and object planes

the image center. As a result, the centers of objects in simulatenously captured focused and defocused images no longer align (Fig. 4.12), and the calibration procedure becomes self-defeating.

While this error is easy to account for in the ideal case of right angles and perfect alignments (simply rescaling the image would solve the problem) the situation becomes more difficult in practice when the target pattern is no longer parallel to the camera sensor (intentionally or accidentally) or when cylindrical lenses are used to add optical compression. In fact, there is no guarantee that affine mappings are sufficient in the general case.

4.4.1 Misidentified droplets in Dantec DynamicStudio software

Table 4.2 shows that the identification of disk images (and particularly overlapping disk images) is, in practice, highly problematic and only rarely leads to reasonable results. Most telling is the sensitivity of the disk recognition algorithm to extremely small changes in the settings (e.g. expected disk image diameter in pixels or windowing type).

The success rate of our own attempts to recognize overlapping disks has been disappointing, as evidenced in the Appendix.

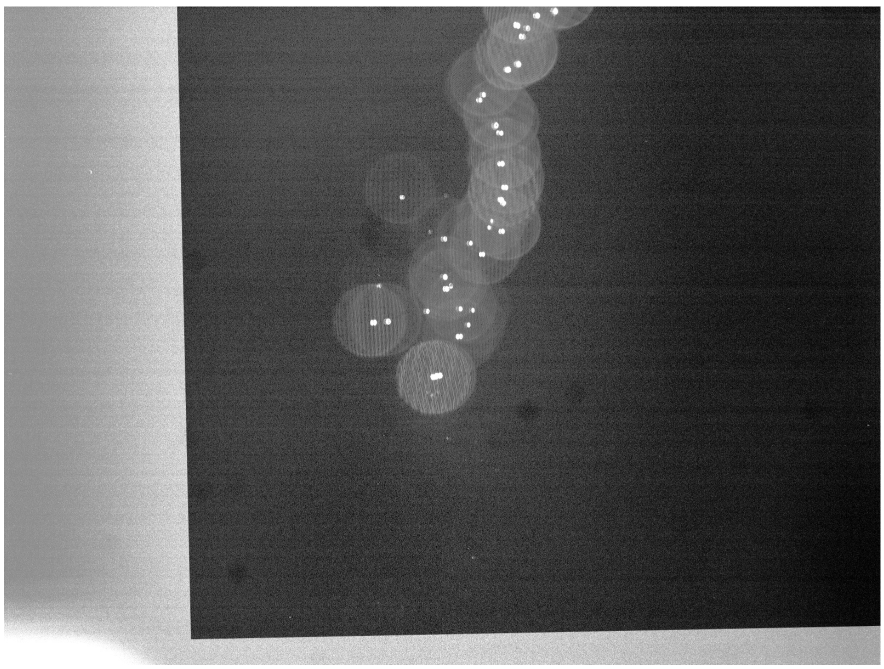


Figure 4.12: Focused camera image, after applying homography **H** derived from the calibration images, is superimposed onto defocused camera image of droplets. Discrepancies between object centers grow towards the edge of the image.

Config	ml/h	Hz	True Dia.	D_{10}	D_{32}	Count	Std. Dev.
A	110.7	1192	366.6	282.5	314.8	95	66.9
B	110.7	1192	366.6	264.0	279.9	95	44.1
C	110.7	1192	366.6	80.3	82.4	126	10.4
D	110.7	1192	366.6	75.6	149.6	279	45.3
E	110.7	1192	366.6	73.9	142.9	275	42.7
A	110.7	1960	310.6	79.6	160.3	629	49.5
B	110.7	1960	310.6	213.4	239.6	167	50.5
C	110.7	1960	310.6	164.9	199.5	488	48.4
A	110.7	1982	309.4	802.9	810.7	227	54.0
B	110.7	1982	309.4	524.6	674.6	495	249.2
C	110.7	1982	309.4	524.6	674.6	495	249.2
A	110.7	2044	306.3	92.8	112.4	284	81.4
B	110.7	2044	306.3	142.0	182.9	216	53.2

Table 4.2: Some sample data showing the effect of changing the image processing parameters in the FlowSizer software ("Config" column) very slightly. The "D_{10}" and "D_{32}" columns list the respective averages found by the FlowSizer software, respectively; the "Count" column lists the number of recognized disks. The "Std. Dev." column indicates the standard deviation in the size distribution found.

4.4.2 Eliminating the centre discrepancy effect

Surprisingly, only Hardalupas *et al.* [41, 42] have hitherto published a discussion of this effect, and the only previous mention known to the authors is in Kurosawa et al. [43], who dismissed it as a "positioning error".

Hardalupas *et al.* identified the centers of particles in both PIV (focused) and ILIDS (defocused) images. They then empirically estimated the magnitude of the center discrepancy effect along the vertical axis, which enabled them to improve the accuracy of their nearest-neighbour-based droplet image matching algorithm.

We found that algorithms developed by the computer vision community in recent years can obviate the need for calibration entirely. Instead, we can use visual correspondences between the focused and defocused images to find the mapping between them directly. To that end, we first provide in Section 4.4.3 a brief overview over popular methods in the field of automated (linear) *registration*, i.e. the art of finding a *homography* (geometric mapping) between two *epipolar images* (images of the same object, taken from different positions and angles). Section 4.4.4 documents our approach in greater detail and shows the result of a successful recalibration. An overview is provided by Fig. 4.13.

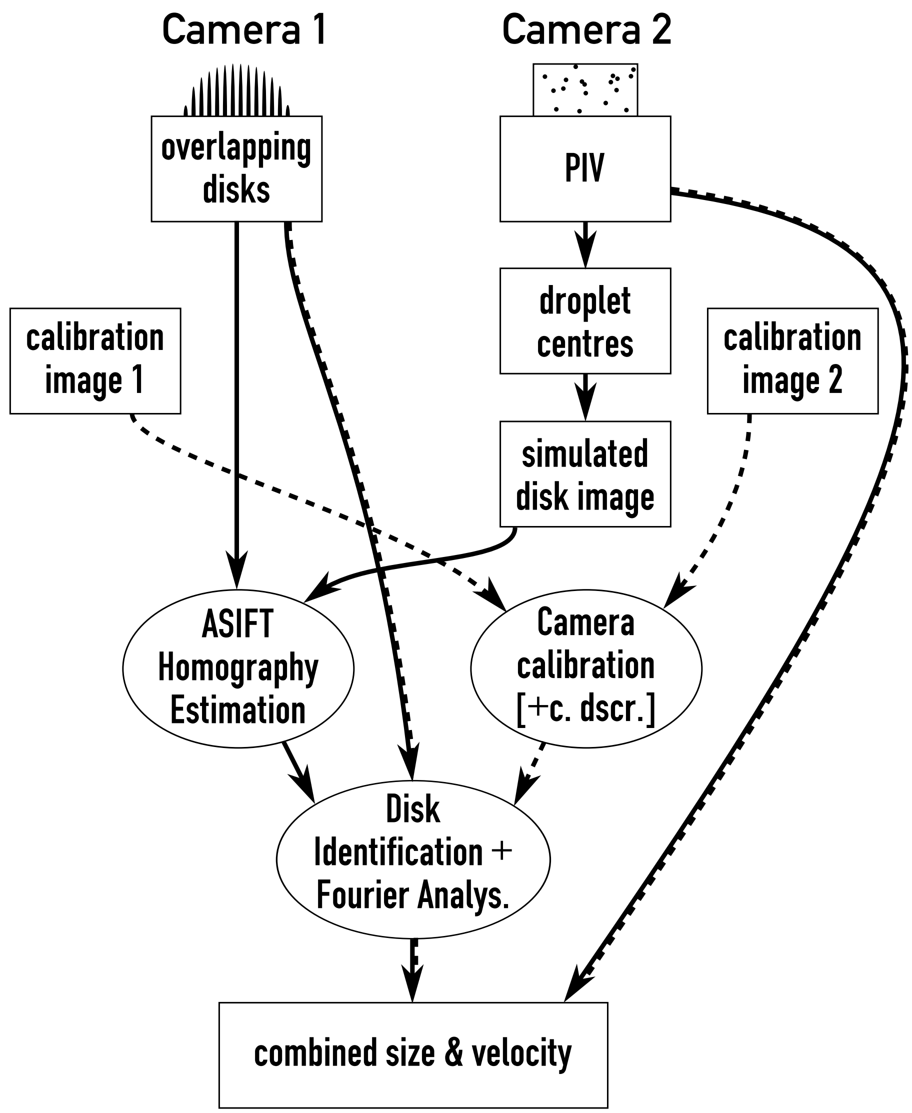


Figure 4.13: Procedural chart outlining how fringe disk images can be combined with PIV images to yield integrated size and velocity information for every droplet. Solid arrows: this paper; dashed arrows: typical approach using camera calibration (e.g. Dantec DynamicStudio software).

4.4.3 Review of image registration techniques

Given two identical images that have been rotated, shifted or even scaled with respect to one another, the applied transformation can theoretically be found by means of a brute-force search. This method is not feasible in practice, not only because of its enormous computational complexity (there are no gradients to guide the search) but also because of its inability to deal with noise, focal blur, perspective changes and other nonlinearities introduced by the photographic process. Conversely, normalized cross-correlation measures between images, as commonly used in PIV, are unaffected by noise but not invariant to rotation and scale and therefore not generally practical. The standard approach to image registration is therefore a three-step process. First, *keypoints*, i.e. "interesting" points in the images are found by a keypoint detection algorithm. Then, a small image patch at every keypoint is extracted and converted into a *feature vector*, a set of numbers providing a very general description of the image patch that accounts for scale, rotation, blur, contrast, etc. Finally, matches between similar feature vectors from the two images are found, outliers are removed, and the homography is calculated.

However, the results of a keypoint detection algorithm must be as repeatable as possible, i.e. the same set keypoints should be found in both images regardless of their relative position, rotation, scale, etc. For instance, the Harris corner detector [44], one of the earliest keypoint detectors, is sensitive to scale and thus often unusable.

The recent decade has seen a rapidly growing collection of proposed keypoint detectors, beginning with SIFT [45], SURF [46] and BRISK [47], all of which include keypoint extractors, to CENSURE [48], optimized for speed, and FAST [49], which incorporates machine learning methods. Finally, the recent publication of ORB [50] includes a rotation-aware version of FAST used in this paper. Many more have been developed but are not included here for brevity's sake.

Keypoint extractors (sometimes called *descriptors*) are often optimized for and therefore included with keypoint detectors, as in the instances mentioned above. Some however are standalone algorithms, such as BRIEF [51].

It is straightforward to find matching keypoints by searching for pairs with the smallest arithmetic distance between their feature vectors (e.g. using the L^2 norm). This nearest-neighbour search can be done exhaustively in linear time to find the optimal matching, but many faster, if approximate, search methods exist. We should note FLANN [52], a publicly available collection of such implementations which includes a fully automatic parameter selection heuristic.

Finally, the homography, assuming one exists, can be derived from the set of matched keypoint coordinate pairs. Since many of the found matches will be wrong, it is of essence to use a robust estimator, i.e. a type of regression model designed to ignore outliers. Possibly the oldest of these methods is RANSAC [53], an iterative procedure in which sets of data points are chosen at random and discarded if the agreement between a model fit to them and all other data points falls below a carefully chosen threshold. RANSAC was used for this paper, although other robust methods exist. The criterion developed by Moisan and Stival [54] deserves special mention in our context; it does away with RANSAC's hard threshold and instead takes into consideration the probability of a match to be in consensus with epipolar geometry.

4.4.4 Using affine oriented FAST, BRIEF and RANSAC to estimate the homography between PIV and ILIDS photographs

Existing PIV/ILIDS systems derive the homography from the result of a camera calibration procedure which the user is required to perform before analyzing images. Although the final value of $\mathbf{H}$ is invisible to the user in our copy of Dantec's DynamicStudio software, the camera matrices $\mathbf{P}_{\mathrm{foc}}$ and $\mathbf{P}_{\mathrm{def}}$ can be shown and edited. We therefore must find a corrected homography $\hat{\mathbf{H}}$ that allows us to compute

$$\hat{\mathbf{P}}_{\mathrm{def}} = \hat{\mathbf{H}}\,\mathbf{P}_{\mathrm{foc}} \tag{4.15}$$

so that we can replace $\mathbf{P}_{\mathrm{def}}$ with $\hat{\mathbf{P}}_{\mathrm{def}}$ in the software, effectively correcting $\mathbf{H}$ to $\hat{\mathbf{H}}$.

To efficiently extract keypoints, we combined three algorithms: ASIFT [55] to deal with skew transformations; an oriented version of FAST, published as part of ORB, to detect keypoints; and standard BRIEF as a keypoint extractor.

ASIFT is a method originally developed to be used with SIFT. It introduces invariance to affine mappings by simulating various projective transformations while FAST and BRIEF are run repeatedly. This slows the analysis down, but given the infinitude of possible angled camera-camera-object configurations, it is wise to maintain a flexible framework.

We should note that the original ASIFT with SIFT works well, but SIFT is encumbered by patents. To encourage vendors of imaging systems to adopt the proposed algorithms, we made it our goal to find a freely available replacement.

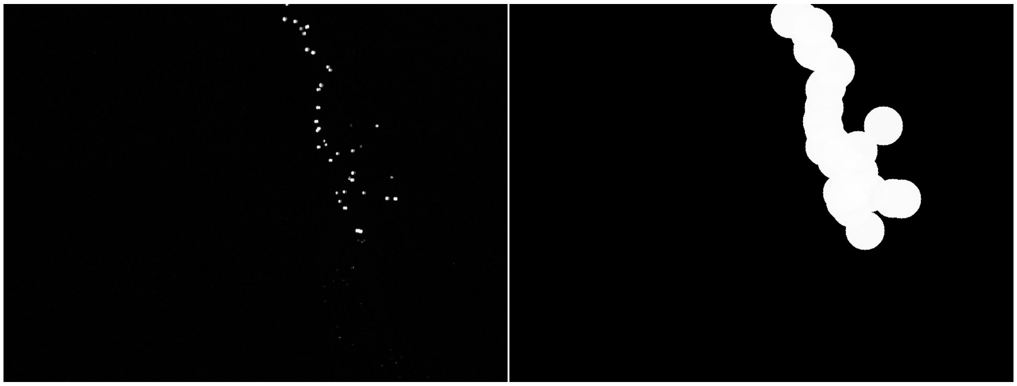


Figure 4.14: Simulating disks based on the focused image.

Recall that the disks in the defocused image are missing from the focused image, rendering a registration between them impossible. It is straightforward to simulate the disks, however. We followed the following protocol on our focused images:

1. Mask the image, blacking out all areas that are known not to contain droplets.
2. Subtract the pixel-wise minimum or mean value taken over all images taken by the camera. This step serves to black out defective hot pixels on the camera's CCD and other static noise.
3. Erode the image, using a 3×3 or 5×5 kernel. This will close any remaining bright pixels which are likely noise.
4. Locate the intensity peaks in the remaining image.
5. Fill a blank image with black, then draw bright circles of diameter D_{disk} onto it, centerd at the respective positions of the intensity peaks detected in the focused image. (Note that simply dilating the result of the previous step will not lead to circular disks.)

The result of performing these operations on our sample image is shown in Fig. 4.14. We determined the disk diameter D_{disk} empirically from the defocused images, although it is naturally preferable to automate this step, e.g. using circular Hough transforms or cross-correlation with circular masks. There may be simpler ways of achieving the same result, e.g. by means of Gaussian filters, distance transforms and thresholding operations. However, we found the protocol described above to be quite robust to noise and fast enough

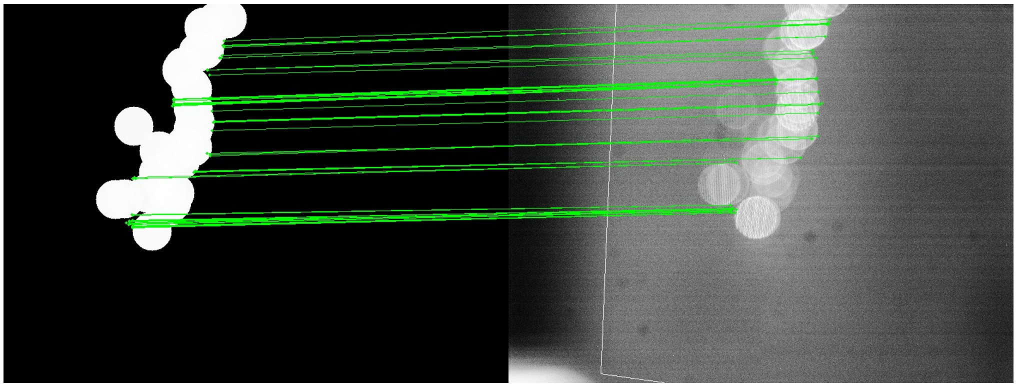


Figure 4.15: Visualized inliers in the set of matched keypoints between the mirrored simulated disks (see Fig. 4.14) and the ILIDS image.

for our application.

Implementations of ORB and BRIEF are freely available through the OpenCV project, which provides bindings for the C++ and Python languages. We used these implementations to find and extract matching keypoints between our sample images, shown in Fig. 4.15.

The matches shown in Fig. 4.15 were found using a most basic method: brute-force match search, followed by a RANSAC estimation of the homography matrix $\mathbf{K}$ using a threshold of 10.

Since the two cameras were positioned behind a beamsplitter in our setup, the defocused image was flipped horizontally. We therefore first mirrored it horizontally, using the transformation matrix

$$\mathbf{M}_h = \begin{bmatrix} -1 & 0 & \text{(image width)} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

To speed up the image registration process, it can be helpful to first down-scale the images. To reduce an image to half of its original size, apply

$$\mathbf{S}_{0.5} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

While the above operations might not be strictly necessary, we found that they significantly improved the quality of the matches identified. If the registration algorithms mentioned above now find a homography matrix $\mathbf{K}$, then we can write

$$\mathbf{K}\,\mathbf{M}_h\,\mathbf{S}_{0.5}\,\mathbf{P}_{\text{foc}} = \mathbf{S}_{0.5}\mathbf{P}_{\text{def}} \tag{4.16}$$

and to bring this into a form similar to (4.14),

$$\begin{aligned}\mathbf{S}_{0.5}^{-1}\,\mathbf{K}\,\mathbf{M}_h\,\mathbf{S}_{0.5}\,\mathbf{P}_{\text{foc}} &= \mathbf{S}_{0.5}^{-1}\,\mathbf{S}_{0.5}\mathbf{P}_{\text{def}} && (4.17)\\ &= \mathbf{P}_{\text{def}} && (4.18)\end{aligned}$$

Finally, it turns out that Dantec's DynamicStudio software violates convention by placing the coordinate origin at the bottom (not top) left corner of the image. We must therefore pre- and post-multiply by $\mathbf{M}_v^{\pm 1}$, with

$$\mathbf{M}_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & (\text{image height}) \\ 0 & 0 & 1 \end{bmatrix},$$

to arrive at our final expression for $\hat{\mathbf{H}}$:

$$\hat{\mathbf{H}} = \mathbf{M}_v\,\mathbf{S}_{0.5}^{-1}\,\mathbf{K}\,\mathbf{M}_h\,\mathbf{S}_{0.5}\,\mathbf{M}_v^{-1}. \tag{4.19}$$

Substitution of $\hat{\mathbf{H}}$ into (4.15) yields $\hat{\mathbf{P}}_{\text{def}}$, which can be manually entered into the DynamicStudio software. Fig. 4.16 illustrates how the use of $\hat{\mathbf{H}}$ leads to an improved alignment compared to Fig. 4.12. Note that a slight projective distortion is necessary for optimal registration, confirming that it is infeasible to restrict the homography to affine matrices.

4.5 Removing center discrepancies with point set registration

The keypoint matching approach described above is not applicable when a slit aperture was used to reduce overlap, as in the paper by Hardalupas *et al.*, so we will outline briefly

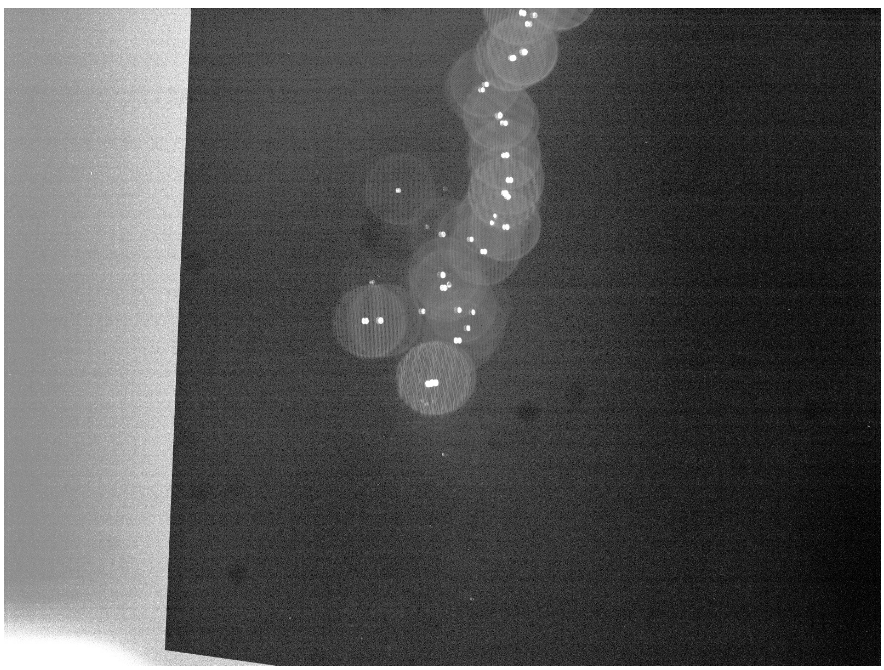


Figure 4.16: Focused camera image, after applying corrected homography $\hat{\mathbf{H}}$ derived from the matched keypoints, is superimposed onto defocused camera image of droplets.

how to use registration algorithms with such setups.[5] A procedural overview is provided by Fig. 4.17.

Keypoints are not required, when the absence of overlap allows us to identify focused and defocused objects centers directly from the respective images and find a projection mapping between them. Indeed, Hardalupas *et al.* successfully registered their PIV and ILIDS images that way: using wavelet transforms at various frequencies, they identified the putative droplet centers on both focused and defocused images. Then, using a continuous, single-stream monodisperse droplet generator, they estimated how the magnitude of the center discrepancies varied over the image. After applying this empirically estimated distortion to the captured focused images, they matched each focused droplet to the closest defocused droplet (if one could be found within an subjectively chosen search distance).

Although they reported good success using this method, it requires both an empirical estimation of the center discrepancies every time the camera is defocused *and* a guess at the appropriate search window size. Moreover, mismatches are likely as the naive closest-neighbour search is not robust to noise. To eliminate these steps, we suggest that droplet matches be found directly using a robust point set registration algorithm.

Since the early 1990s, computer vision researchers have accumulated an impressive body of work on this topic, most of it focusing either on rigid transformations (i.e. translation and rotation only) or non-rigid transformations (typically understood to include nonlinear warping). The problem at hand requires an algorithm able to deal with projective transforms, which are non-rigid but linear.

The only paper known to the authors to specifically address this case is by Chi et al. [56], who propose an iterative search based on image moments. Since image moments are an aggregate metric, they do not directly lead to a droplet-to-droplet correspondence. Still, closest-neighbour matches after application of this algorithm would likely produce results no worse than those found after estimating the transformation empirically.

Robust non-rigid methods are also applicable in this case and deserve some mention. Many of them are probabilistic relaxations of the Iterative Closest Point algorithm, which simply searches for the least-squares-optimal rigid mapping. Several of these approaches were reviewed and generalized by Jian and Vemuri [57]. A slightly different approach, named Coherent Point Drift[58], is also highly popular and illustrated in Fig. 4.18.

We forgo at this point a documentation of the application and refer the reader to Hardalu-

[5] While slit strip images could be simulated over the focused image (in a procedure analogous to that illustrated in Fig. 4.14), the lack of overlap between them could make it significantly more difficult to find “interesting” keypoints in the simulated image.

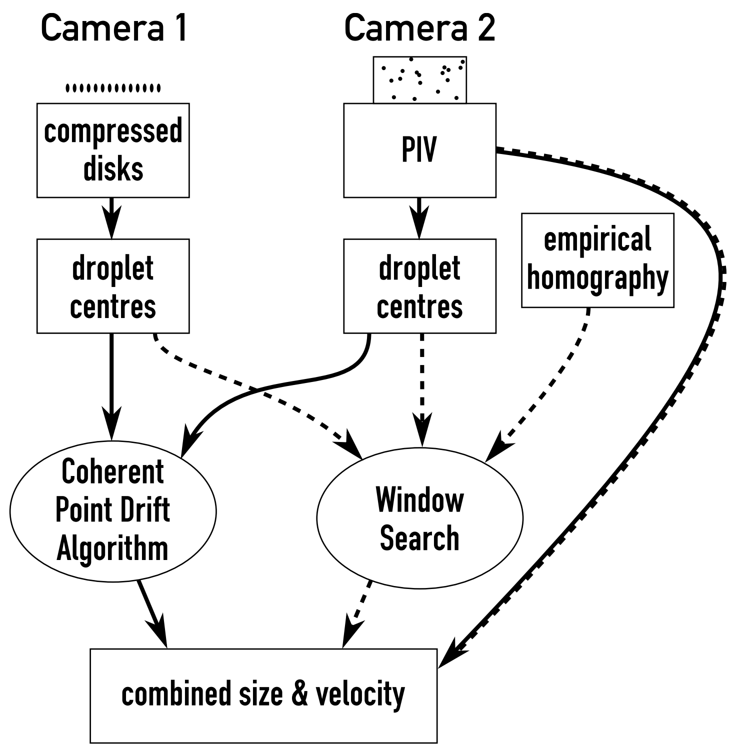


Figure 4.17: Procedural chart outlining how compressed fringe disk images can be combined with PIV images to yield integrated size and velocity information for every droplet. Solid arrows: this paper; dashed arrows: approach taken by Hardalupas et al. [41].

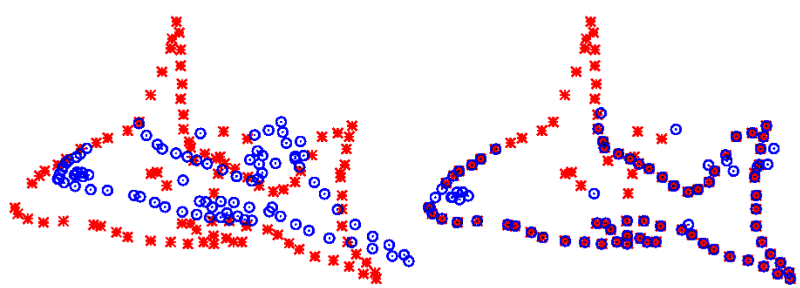


Figure 4.18: Non-rigid variant of the Coherent Point Drift algorithm applied to two point sets. Notice that the probabilistic nature of the matching creates robustness to unmatched points. (Image source: Wikipedia)

pas *et al.*, who describe their center identification technique in good detail, and to the above-mentioned authors, who have published freely available implementations of their algorithms online.

Phase-Doppler Particle Analysis (PDPA)

Phase-Doppler anemometry and droplet sizing techniques are based on the far-field intensity fluctuations in the laser light scattered by passing droplets. Unlike ILIDS, PDPA uses intersecting laser beams (instead of a laser sheet) to illuminate the droplets. Whereas ILIDS *images* the infringement pattern cast by the glare points on the illuminated droplet, PDPA measures the fringe spacings indirectly by relating them to the phase difference between the signals recorded by a pair of adjacent detectors.

Doppler [59] suggested in 1842 that the slight differences in wavelength between the colours of various stars could be used to determine the stars' velocities relative to Earth. A century later, military radar operators during World War II realized that the Doppler effect could be exploited to estimate target velocities using their radar systems. Soon, meteorologists had adopted the technology to measure wind speeds [60]. The flurry of activity around Doppler measurements in the ensuing years produced the first laser-based particle anemometry system, designed in 1964 by Yeh and Cummins [61], and several novel beam-detector configurations have been proposed since. Among them are the popular dual-beam configurations, constitutiting a departure from the original reference beam systems which can be more difficult to align. A exemplary dual-beam setup is shown in Fig. 5.1.

Laser-Doppler anemometry (LDA, also known as laser-Doppler velocimetry or LDV) provides velocity measurements only. The 1970s saw the discovery that the fringe patterns cast by the scattered light could be analyzed to yield size information [62, 63]. This technique is now widely used and often integrated into the detecting and processing hardware of commercial LDA systems, typically under the name of phase-Doppler Anemome-

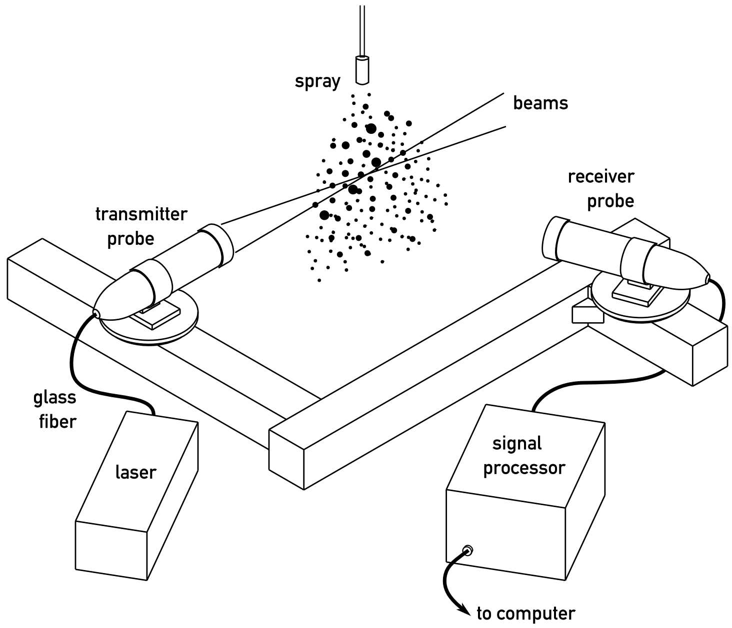


Figure 5.1: Typical commercial PDPA configuration using transmitter and receiver probes

try (PDA) or phase-Doppler particle analysis (PDPA).

PDPA can be used to ascertain droplet size distributions with a very high precision. Nevertheless, its accuracy hinges on the knowledge of a large number of distances, angles and voltages. A single erroneous one of them will throw off the result, leaving the operator none the wiser. We shall therefore discuss briefly the various sources of error in this chapter, and provide some pointers to their resolution. The focus is on measurements of size, not velocity.

5.1 Optical principle

Only droplets passing through the measurement volume have a chance of being detected and characterized, so we will provide here a description of the measurement volume and its interaction with a falling droplet.

To create the measurement volume, two laser beams must intersect where their wave fronts are planar. To this end, the transmitter probe features a front lens which focuses the beams to a small but finite *waist diameter*. Note that focusing onto an infinitesimally small point is impossible: as the beam is exiting from a finite-sized aperture, some degree of diffraction is unavoidable and will lead to a finite diameter. In the case of most commercial setups, the laser exhibits a Gaussian intensity profile, i.e. the light intensity falls off smoothly with increasing distance from the beam axis. The diffraction of a Gaussian beam generally causes spherically expanding wavefronts [64]; the focusing lens counteracts this effect and leads to plane waves at the beam waist.

Borrowing from the terminology used by Albrecht et al. [1], the *measurement volume* is the space in which a particle must be located such that it scatters the light onto the detector. For large particles, this will not coincide perfectly with the illuminated volume. The measurement volume contains the *detection volume*, within which a droplet scatters enough light onto the detector to exceed the chosen minimum threshold for detection.

In accord with most introductory texts on PDPA, we will offer two descriptions of the scattering effect experienced by the droplets falling through the measurement volume: a fringe model for very small droplets, and a Moiré model for larger droplets.

5.1.1 Particles smaller than the wavelength

The intersection of the two beams results in their interference at an angle Θ. The alternating extinction and amplification creates a pattern of fringes. As the small droplet (assumed to be a single point) falls through the fringes, it samples the light intensity, scattering an alternating signal onto the detector, which registers a burst of voltage spikes. Since the fringe spacing is known and constant, the detected burst frequency is linearly related to the velocity of the falling droplet. The measurement of droplet size is identical for both small and large droplets and explained below.

5.1.2 Large particles

In practice, we do not deal with droplets smaller than tens of microns (much less submicron particles, to which the above would apply). We therefore apply a somewhat more complex model, in which the fields of both scattered beams are considered separately. Indeed, their interference is modelled directly at the receiver.

Depending on the the relative refractive index m, either the reflected light or the light of a certain scattering order dominates when seen from an off-axis angle φ. In the setup shown in Fig. 5.1, $\varphi = 60^\circ$ and m is assumed to be 1.33 (water droplets in air). In this setup,

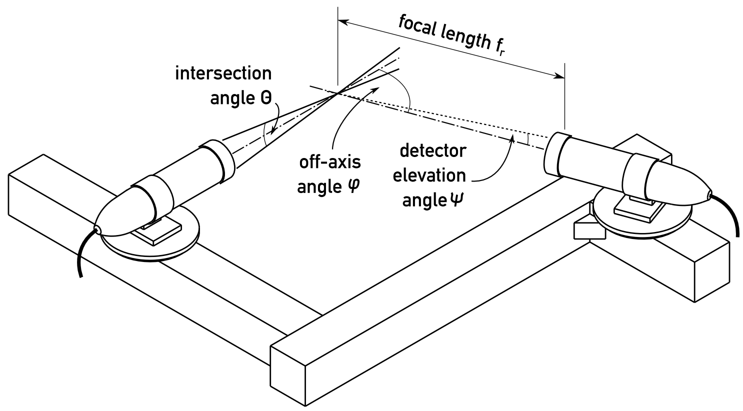


Figure 5.2: Nomenclature used for the geometry in Fig. 5.1

the receiver will see the beams mostly as first-order refracted rays. An extensive numerical evaluation of these relationships was published by Naqwi and Menon [11].

From the perspective of the receiver, each beam is scattered from a glare point on the droplet surface. Due to the path length differences and the falling motion past the receiver probe, the intensities from both glare points create a beating frequency on both detectors. The detectors are offset in space (by a distance on the order of millimeters) and therefore measure the same beating signal with a slight phase shift (on the order of a few π).

Fig. 5.3 illustrates how larger droplets result in narrower fringes, and thus a larger phase difference. Note, however, that the patterns shown in Fig. 5.3 are valid only for stationary droplets. Falling droplets experience a Doppler effect: the upward-angled beam is sampled at a relatively higher velocity, leading to a higher scattered frequency. The downward-angled beam, on the other hand, is scattered at a lowered frequency. The discrepancy in frequency leads to curved fringes not shown in Fig. 5.3. Moreover, the direction of the receiver changes as the droplet is falling. The result is not directly intuitive: the velocity of the droplet is cancelled by the curving of the fringes, so it has no impact on the phase difference measured between detectors A and B. Similarly, the size of the droplet has effectively no influence on the beat frequency (i.e. burst frequency) that is used to derive velocity information.

For first-order refraction, which according to Naqwi and Menon [11] is the dominant scattering mechanism for water droplets in air seen from $\varphi \approx 60°$, Albrecht et al. [1] provide the following relationship between droplet diameter D_p and the difference in mea-

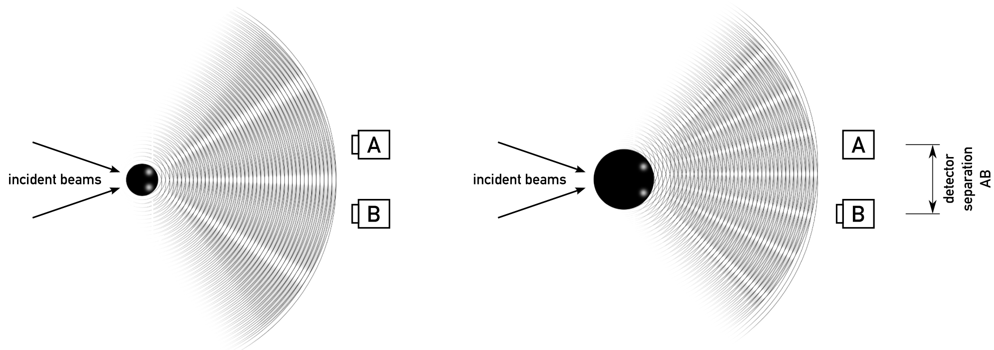


Figure 5.3: Moiré-style visualization of the interference pattern cast by the two glare points. Left: glare points on small particles are close, resulting in wide fringes. The signals recorded by detectors A and B are shifted in time by a phase difference $\Delta\Phi_{AB}$, which is due to the detectors' known separation in space. Right: glare points on larger particles are farther apart, resulting in narrower fringes and a larger phase difference.

sured phase Φ between two detectors A and B:

$$
\begin{aligned}
\Delta\Phi_{AB} &= \beta_{AB} D_d \\
&= \frac{2\pi}{\lambda}\left(\Phi_A - \Phi_B\right) D_d \\
&= \frac{2\pi}{\lambda} D_d \Bigg[\sqrt{1 + m^2 - \sqrt{2}m\sqrt{1 + \sin\psi_A \sin\frac{\Theta}{2} + \cos\psi_A \cos\varphi_A \cos\frac{\Theta}{2}}} \\
&\quad - \sqrt{1 + m^2 - \sqrt{2}m\sqrt{1 - \sin\psi_A \sin\frac{\Theta}{2} + \cos\psi_A \cos\varphi_A \cos\frac{\Theta}{2}}} \\
&\quad - \sqrt{1 + m^2 - \sqrt{2}m\sqrt{1 + \sin\psi_B \sin\frac{\Theta}{2} + \cos\psi_B \cos\varphi_B \cos\frac{\Theta}{2}}} \\
&\quad + \sqrt{1 + m^2 - \sqrt{2}m\sqrt{1 - \sin\psi_B \sin\frac{\Theta}{2} + \cos\psi_B \cos\varphi_B \cos\frac{\Theta}{2}}} \Bigg].
\end{aligned}
\tag{5.1}
$$

Note that Fig. 5.2 explains some of the notation used above.

Because of the periodic nature of the arriving signal, any phase difference value exceeding one period (2π) is indistinguishable from one below one period. One solution to this problem is to consider the signal bursts from both detectors in their entirety and to determine the time shift between them (time shift technique) [1]. Another, more commonly

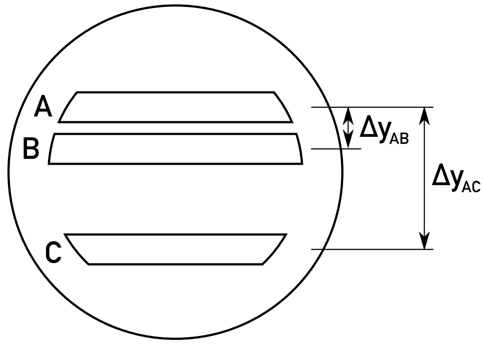


Figure 5.4: Typical vertical arrangement of detectors A, B, and C in a receiver probe. Detector separation values are indicated.

implemented solution is the addition of a third detector, spaced very closely to the first. Then the phase difference between detectors A and C is

$$\Delta\Phi_{AC} = \beta_{AC} D_d. \tag{5.2}$$

Since the slopes β_{AB} and β_{AC} are known, we can then deduce how many 2π jumps should be contained in the measured $\Delta\Phi_{AB}$:

$$n_{2\pi} = \mathrm{int}\left[\frac{1}{2\pi}\left(\frac{\beta_{AB}}{\beta_{AC}}\Delta\Phi_{AC} - \Delta\Phi_{AB}\right) + \frac{1}{2}\right] \tag{5.3}$$

This allows us to bring $\Delta\Phi_{AB}$ into the range $[0, 2\pi]$ by subtracting the redudant $n_{2\pi}$ periods:

$$\Delta\Phi_{AB} = \beta_{AB} D_d - 2\pi n_{2\pi}, \tag{5.4}$$

and are able to solve for D_d:

$$D_{d,AB} = \frac{\Delta\Phi_{AB} + 2\pi n_{2\pi}}{\beta_{AB}} \tag{5.5}$$

$$D_{d,AC} = \frac{\Delta\Phi_{AC}}{\beta_{AC}} \tag{5.6}$$

If the discrepancy between the two values is below a set threshold, the measurement is accepted.

Generally, the elevation angle ψ of a given detector is not known a priori, at least not to sufficient accuracy. However, it can be estimated from the detector separation values (see Fig. 5.4) and the focal length of the receiver's lens f_r. The detector separation values, in turn, are best found via a calibration procedure as outlined below.

5.1.3 Calculation of droplet sizes in the TSI FlowSizer software

To verify the results displayed by the TSI FlowSizer software, it was necessary to understand how FlowSizer arrives at a diameter distribution from the measured phase data. Although their algorithm is not published, it is possible to reverse engineer their approach by deriving the required mathematical simplifications. We then compared the results to an implementation of the full equation given above, based on a set of raw detector phase data.

FlowSizer records the phase differences between detector pairs AB and AC, ranging from $-180°$ to $180°$. We must first adjust these values to yield strictly positive phase differences. To this end, we add 360° to all negative phase difference values to map them onto equivalent phositive phase differences.

Albrecht et al. [1] then provide a simple approximation of (5.1) under a number of assumptions (e.g. $\sin\psi \approx \psi$ etc.):

$$\Delta\Phi_{AB} \approx -\frac{2\pi}{\lambda} d_p \sin\psi_{AB} \sin\frac{\Theta}{2} \frac{m}{\nu\sqrt{1+m^2-m\nu}} \tag{5.7}$$

$$\Delta\Phi_{AC} \approx -\frac{2\pi}{\lambda} d_p \sin\psi_{AC} \sin\frac{\Theta}{2} \frac{m}{\nu\sqrt{1+m^2-m\nu}}, \tag{5.8}$$

with

$$\nu_{AB} = \sqrt{2(1+\cos\psi_{AB}\cos\varphi\cos\frac{\Theta}{2})} \tag{5.9}$$

$$\nu_{AC} = \sqrt{2(1+\cos\psi_{AC}\cos\varphi\cos\frac{\Theta}{2})} \tag{5.10}$$

FlowSizer first makes the assumption that $\cos\psi_{AB} \approx \cos\psi_{AC} \approx 0.995$, and thus simply computes

$$\nu = \sqrt{2(1+0.995\cos\varphi\cos\frac{\Theta}{2})} \tag{5.11}$$

They also designate as "slope" (here, γ) the second term of (5.7) and (5.8):

$$\gamma = \frac{m}{\nu\sqrt{1+m^2} - m\nu}. \tag{5.12}$$

Also, the fringe distance at the center of the illuminated volume, given perfect alignment, is given by Albrecht et al. [1] to be:

$$\delta x = \frac{\lambda}{2\sin\frac{\Theta}{2}}. \tag{5.13}$$

Finally, FlowSizer appears to use the approximation

$$\sin\psi_{AB} = \frac{\Delta y_{AB}/2}{f_r} \tag{5.14}$$

$$\sin\psi_{AC} = \frac{\Delta y_{AC}/2}{f_r}, \tag{5.15}$$

where Δy_{AB} is the physical separation between detectors A and B (e.g. in millimeters) and f_r is the focal length of the receiver lens.

It follows that (5.7) and (5.8), when rewritten for β, can be composed as

$$\beta_{AB} = \frac{\Delta y_{AB}}{2f_r}\gamma\frac{360°}{\delta x} \tag{5.16}$$

$$\beta_{AC} = \frac{\Delta y_{AC}}{2f_r}\gamma\frac{360°}{\delta x} \tag{5.17}$$

Where we write 360° instead of 2π since FlowSizer works with phase values in degrees. This is a useful approximation, because now β_{AB} and β_{AC} only depend on constant values (including Δy_{AB} and Δy_{AC}, which can be set in the software).

FlowSizer then filters out all $d_{p,AB}, d_{p,AC}$ pairs that differ by more than some threshold. This is referred to as the "epsilon cutoff" in the TSI FlowSizer software. The manual recommends seven percent of the maximum measureable size, which in practice evaluates to a few tens of microns.

5.1.4 Calibration in TSI FlowSizer

It follows from the above equations that no calibration should be necessary at all, as long as all angles and distances are well-known. However, Δy_{AB} and Δy_{AC} are not typically known to great precision. The solution, then, is to determine their values based on the phase differences recorded after measuring monodisperse droplets of known size.

The TSI FlowSizer software provides this functionality. Presuming a vibrating orifice droplet generator is used, flow rate and frequency can be entered and are converted to droplet diameter using (3.2). The software then adjusts Δy_{AB} and Δy_{AC} such that the peak (if any) or some still-unknown measure of average of the diameter distribution aligns with the expected droplet diameter. This is easily done, since due to the approximation of $\cos\psi \approx 0.995$, the value of γ no longer depends on Δy_{AB} and Δy_{AC} and can be set independently.

5.2 Sources of error

In this section, we provide an extensive review of possible sources of error in the PDPA measurement of droplet sizes. It may serve as a troubleshooting guide and it provides context for Section 5.3.

5.2.1 Gaussian beam divergence

The theory predicting the linear relationship between detector phase difference and droplet diameter is founded on the assumption of very small droplets and plane beam wavefronts. Unfortunately, when particles larger than the beam cross the measurement volume along certain paths, glare points of the normally dominant scattering order (e.g. first-order refraction) can temporarily become weak enough to be dominated by another scattering order (e.g. reflection). This phenomenon leads to erroneous results, of course, and is termed *trajectory ambiguity effect* (TAE) or *Gaussian beam defect* (GBD) in the literature.

The problem was recognized first by Saffman [65], but had to be neglected until the plane-wave scattering theory of Lorenz-Mie optics was extended into the *Generalized Lorenz-Mie Theory* (GLMT) which encompass incident wave fronts of all shapes. Most of this work was done by Gouesbet, Gréhan, Maheu, Lock and others throughout the 1980s [66, 67, 68, 69], and mathematically rigorous formulations were available by the early 1990s [70, 71]. The 1996 paper by Gouesbet and Gréhan summarizes these developments and provides an early overview over the attempts to circumvent the TAE [72].

Solutions proposed in Albrecht et al. [1] include planar setups (a different geometry, in which detectors are placed in line with the beams) and validity checks between the diameter determined by detector pairs AB and AC.

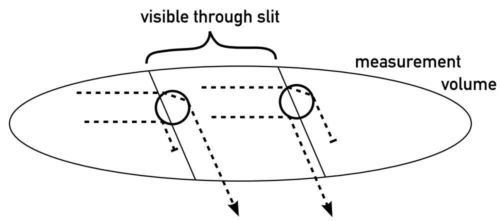


Figure 5.5: Schematic illustrating the slit effect. Two small droplets are passing through the measurement volume (top view; trajectories are into the page). The droplet on the left only scatters its refracted glare point onto the receiver (as normally intended at $\varphi = 60°$). Refraction is suppressed, however, for the droplet on the right; the slit aperture only allows the reflected light to pass.

5.2.2 Slit effect

Most commercial phase-Doppler systems use a slit aperture in front of the detector or receiving fibers [1], to help clearly define the detection volume in the z dimension (i.e. the beam direction). This accomplished two things: first, it makes possible the calculation of flux statistics (i.e. droplets per volume and time) and, second, it can be helpful when the spray are very dense, as multiple droplets may be present within the measurement volume at a time.

It can occur, however, that the aperture cuts off the dominant-order glare point of a droplet that is just halfway outside of the visible region. In that case, the glare point corresponding to the non-dominant scattering order will dominate and result in a wrong measurement. A graphical representation is shown in Fig. 5.5.

Durst et al. [73] showed experimentally that the slit effect is even more crucial than the Gaussian beam effect. Together with the TAE, this is called the measurement volume effect. As with the slit effect, three-detector setups are significantly less sensitive to this problem because of the additional validation. Qiu and Hsu propose using four detectors, instead of three, to sidestep this problem entirely [74]. A similar four-detector design was verified experimentally by Sipperley and Bachalo [75].

A more recent review of the phenomenon and associated techniques was published by Strakey et al. [76, 77].

5.2.3 Change in fringe frequency over z

The sphericity of the beam wavefronts causes the fringe spacing to vary along the major axis of the measurement volume. At the near and far ends of the volume, the fringes are spaced farther apart, which can result in an error on the order of 10% according to Albrecht et al. [1]. The most important factor is the beam waist dislocation (i.e. misalignment), which must be minimized. The error is also reduced for wider beam waists and larger intersection angles Θ, which in practice translates to setups able to measure very small droplets.

The only way to reduce this error is the proper alignment of laser beams, which can be difficult. The beams can be aligned to intersect by turning small screws inside the transmitter casing while observing their images cast through a microscope objective placed at the measurement point. This method is recommended in the operator manuals. Whether they intersect precisely at their waists, however, is very difficult to determine by this method.

5.2.4 Selection of lenses and masks

Davis and Disimile [78] have shown that the selection of lenses and detector masks can have a considerable effect on droplet sizing results. In their experiment, they measured the same spray using several combinations of focal lengths and mask sizes and found that if the diameter range to be measured is not known a priori, it can be very difficult to choose an appropriate lens-mask combination.

5.2.5 Optical aberrations

Dressler and Kraemer [30] investigated several error sources arising in PDPA calibration, and found that the spherical aberration of the transmitter lens typically gave errors of less than 3%.

5.2.6 Dirty fiber ends

The ends of the transmitter fibers should be polished regularly. Dirty or scratched fiber ends can mar the Gaussian beam profiles, leading to distortions or irregularities in the measurement volume's fringe pattern.

5.2.7 Wrongly entered parameters

The optical principle behind PDPA involves many more geometric parameters than are needed for the operation of e.g. ILIDS or laser diffraction devices. As a result, an excellent user interface and a very attentive user are required to ensure accurate results.

5.3 Calibration of the PDPA sizing system

If all optical parameters are known, the equations necessary to derive droplet diameters and velocities are fully determined. However, Δy_{AB} and Δy_{AC} are not always known to high accuracy and can, in fact, undergo slight changes if the receiver is subject to mechanical vibrations over the years. The FlowSizer software therefore offers a calibration tool to help determine empirical values for Δy_{AB} and Δy_{AC} given a set of measurements of droplets of known size.

To calibrate, we must solve for the two values under the requirement that the peak of the diameter histogram coincide with the known droplet diameter. This is not straightforward for a few reasons:

— The epsilson cutoff criterion (see Section 5.1.3) means that obtaining the diameter histogram from the phase differences is a one-way calculation. Cutoffs cannot be reversed algebraically.

— The meaning of "peak" diameter is ambivalent. If the measured diameter distribution is not exactly symmetrical, it may be preferable to use a form of average (e.g. D_{10}, D_{20}, D_{32}) instead of the diameter corresponding to the true maximum. Alternatively, a Gaussian kernel density estimate of some arbitrary bandwidth may be used. Although a monodisperse droplet stream should result in a very sharp and unambiguous peak, this does not hold in practice in our experience.

— Since there is an infinitude of values pairs for Δy_{AB} and Δy_{AC} that will yield the desired known peak diameter, an additional relationship must be used to establish the best pair. One source for this information could be the difference-diameter plot: we could choose the values such as to minimize the sum of absolute or squared differences between AB and AC measurements.

The combination of the above three sticking points means that an iterative search for the optimal $\Delta y_{AB}/\Delta y_{AC}$ combination is inevitable.

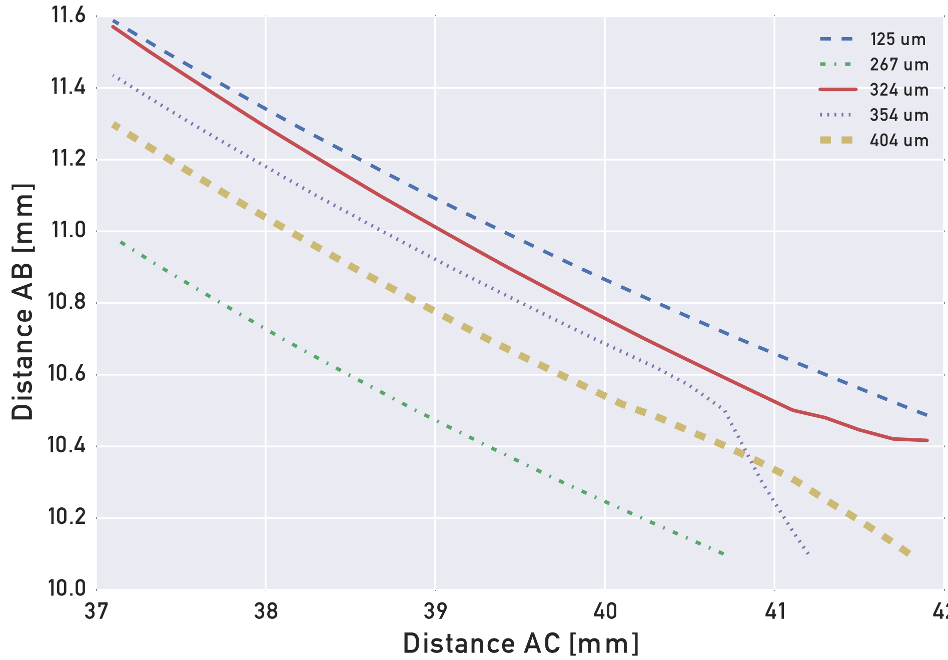


Figure 5.6: Result of a set of calibration attempts showing the difficulty in finding a consistent pair of Δy_{AB} and Δy_{AC} values. Each line corresponds to a set of potential values for Δy_{AB} and Δy_{AC} that would yield a diameter distribution with a peak at the known diameter value shown in the legend.

Alternatively, relaxing the last requirement, we can find the set of possible combinations given one set of measurements, then find overlaps with the sets of possible combinations from other measurement sets.

We have not succeeded in doing so. Fig. 5.6 illustrates the problem: for every calibration attempt using a different droplet size (all of which were verified photographically *and* algebraically via Eq. (3.2)), a different set of $\Delta y_{AB}/\Delta y_{AC}$ values was found that would result in the peak of the measured distribution matching the known diameter. Although Fig. 5.6 shows the result for strict peak values (i.e. argmax), the results are very similar for averages and/or peaks of Gaussian kernel density estimates.

This result, surprising at first, is reflected by the complete unpredictability of the Δy_{AB} and Δy_{AC} values suggested by the FlowSizer calibration tool when given a measured distribution and known diameter (furthermore, it appears that FlowSizer uses the D_{20} measure as the “peak”, which often exceeds the real peak).

Several conclusions may be drawn from this experience:

1. The beams don’t meet at the waist, and the monodisperse droplet stream falls through the measurement volume at a slightly different location with every calibration attempt. The device needs factory recalibration.

2. Other errors are more significant than advertised by the manufacturer, particularly the slit effect.

Most importantly, however, this illustrates that calibrating a PDPA device regularly is crucial—even if it only serves to identify a problem. It is noteworthy that device used had been factory-calibrated twice in the past. Each time, values for Δy_{AB} and Δy_{AC} were found that were well within the ranges shown in Fig. 5.6, but different enough to produce potential errors on the order of 20%.

Conclusions

The research outlined in this thesis led to several conclusions:

1. In practice, the construction of variable-diameter monodisperse droplet generators, particularly on-demand generators, requires very high precision machining work and the reliable absence of both air bubbles and water contaminants. The number of publications on droplet generators belies the difficulty of making and operating one for research purposes.

2. The theory behind ILIDS is sound—we were able to reproduce it using our own software implementation of the necessary image processing algorithms. However, the optical input parameters cannot simply be determined by means of a measuring tape. A proper calibration is necessary; to convert the resulting factor of proportionality into the input values required by the commercial software, an application of the underlying formulae is necessary.

3. The commonly used methods of detecting overlapping disk and/or of combining PIV/LIF and ILIDS imagery are needlessly complex and likely to introduce centre discrepancy errors that have seen almost no treatment in the literature so far.

4. The accuracy of PDPA measurements hinges on the quality of alignment between the intersecting laser beams, which is nearly impossible to ascertain using standard equipment. Since a miscalibrated PDPA will output seemingly perfect (if offset) results, it is crucial that PDPA devices be calibrated regularly.

6.1 Other contributions

The research outlined in this thesis has produced two improvements on existing technologies:

1. A simple and cheap type of continuous-stream droplet generator based on the rotary actuator used in magnetic computer hard drives. The droplet generators work as reliably as the speaker-based ones often used for prototyping, but are much more compact and make no noise.

2. An application of existing image and/or point registration algorithms to the automated alignment of images in double-camera ILIDS setups.

Both findings have been sent out for publication.

6.2 Future work

Future work is necessary to confirm that the difficulties with our PDPA system, described in Section 5.3, are indeed due to misaligned beams and/or dirty fibers. A factory recalibration may be the only way to answer this question with any certainty.

Bibliography

[1] H.-E. Albrecht, M. Borys, N. Damaschke, and C. Tropea. *Laser Doppler and Phase Doppler Measurement Techniques*. Springer, 2003.

[2] W. J. Stry. *The combustion of free flowing droplets of chlorinated benzenes, alkanes, and their mixtures*. PhD thesis, SUNY Buffalo, September 1992.

[3] C. Inoue, T. Watanabe, and T. Himeno. Study on atomization process of liquid sheet formed by impinging jets. In *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, pages 1–14, July 2008.

[4] P. F. Mehegan, D. T. Campbell, and C. H. Scheuerman. Investigation of gas-augmented injectors. Technical Report CR-72703 R-8361, NASA, 1971.

[5] P. Fauchais. Understanding plasma spraying. *Journal of Physics D: Applied Physics*, 37 (9):R86, 2004.

[6] S. P. Newman. Principles of metered-dose inhaler design. *Respiratory Care*, 50(9): 1177–1190, 2005.

[7] I. Endo, Y. Sato, S. Saito, T. Nakagiri, and S. Ohno. Bubble jet recording method and apparatus in which a heating element generates bubbles in a liquid flow path to project droplets, February 2 1988. US Patent 4,723,129.

[8] E. Hecht. *Optics*. Addison-Wesley Longman, Incorporated, 2002. ISBN 9780805385663. URL `http://books.google.ca/books?id=7aG6QgAACAAJ`.

[9] H. van de Hulst. *Light Scattering by Small Particles*. Dover Books on Physics. Dover Publications, 2012.

[10] L. N. Ng. *Manipulation of Particles On Optical Waveguides*. PhD thesis, University of Southampton, September 2000.

[11] A. Naqwi and R. Menon. A rigorous procedure for design and response determination of phase doppler systems. In *Proceedings of the 7th International Symposium on Applications of Laser Anemometry to Fluid Mechanics, Lisbon, Portugal*, pages 340–353, 1994.

[12] W. H. Walton and W. C. Prewett. The production of sprays and mists of uniform drop size by means of spinning disc type sprayers. *Proceedings of the Physical Society B*, 62: 341–350, Jun 1949.

[13] C. R. Merritt and J. A. Drinkwater. A laboratory device for producing single drops in uniform sizes of 150–450 μm diameter. *Pesticide Science*, 8:457–562, 1977.

[14] C. R. Allan, H. R. Carlon, A. K. Stuempfle, T. E. Hoffer, and R. L. Pitter. Survey of the literature: Controlled generation of liquid droplets. Technical Report TR-88164, Chemical Research, Development & Engineering Center, U.S. Army Munitions Chemical Command, Aug 1988.

[15] W. R. Lane. A microburette for producing small liquid drops of known size. *Journal of Scientific Instruments*, 24:98, Apr 1947.

[16] K. Reil and J. Hallett. An apparatus for the production of uniform sized water drops, at desired time intervals. *Review of Scientific Instruments*, 40:533–534, Apr 1969.

[17] S. Coggins and E. A. Baker. Micro sprayers for the laboratory application of pesticides. *Annals of Applied Biology*, 102:149–154, Feb 1983.

[18] A. V. Sergeyev and R. A. Shaw. An inexpensive uniform-size aerosol generator. *Measurement Science and Technology*, 17:N41, 2006.

[19] J. C. Yang, W. Chien, M. King, and W. L. Grosshandler. A simple piezoelectric droplet generator. *Experiments in Fluids*, 23:445–447, 1997.

[20] H. Ulmke, T. Wriedt, and K. Bauckhage. Piezoelektrischer Tropfengenerator zur Kalibrierung von Partikelzählern. *Chemie Ingenieur Technik*, 71(12):1387–1391, 1999.

[21] H. P. Le. *Progress and trends in inkjet printing technology*, volume 42.

[22] E. Lee. *Microdrop Generation*. Nano- and Microscience, Engineering, Technology and Medicine. Taylor & Francis, 2002.

[23] A. Amirzadeh Goghari and S. Chandra. Producing droplets smaller than the nozzle diameter by using a pneumatic drop-on-demand droplet generator. *Experiments in Fluids*, 44:105–114, Jan 2008.

[24] G. J. Green, D. E. Walsh, F. Takahashi, and F. L. Dryer. Aerodynamic device for generating mono-disperse fuel droplets. *Review of Scientific Instruments (ISSN 0034-6748)*, 60:646–652, Apr 1989.

[25] F. Savart. Memoire sur la constitution des veines liquides lancees par des orifices circulaires en mince paroi. *Annales de Chimie et de Physique*, 53:337–386, 1833.

[26] L. Rayleigh. On the capillary phenomena of jets. *Proceedings of the Royal Society of London*, 29:71–97, 1879.

[27] R. J. Donnelly and W. Glaberson. Experiments on the capillary instability of a liquid jet. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 290(1423):547–556, 1966. doi: 10.1098/rspa.1966.0069.

[28] J. M. Schneider and C. D. Hendricks. Source of uniform-sized liquid droplets. *Review of Scientific Instruments*, 35:1349–1350, Oct 1964.

[29] R. N. Berglund and B. Y. H. Liu. Generation of monodisperse aerosol standards. *Environmental Science and Technology*, 7:147–153, Feb 1973.

[30] J. L. Dressler and G. O. Kraemer. A multiple drop-size drop generator for calibration of a phase-doppler particle analyzer. *Liquid Particle Size Measurement Techniques*, 2 (1083):30–44, 1990.

[31] L. Ström. The generation of monodisperse aerosols by means of a disintegrated jet of liquid. *Review of Scientific Instruments*, 40:778–782, Jun 1969.

[32] G. König, K. Anders, and A. Frohn. A new light-scattering technique to measure the diameter of periodically generated moving droplets. *Journal of Aerosol Science*, 17: 157–167, 1986.

[33] A. R. Glover, S. M. Skippon, and R. D. Boyle. Interferometric laser imaging for droplet sizing: a method for droplet-size measurement in sparse spray systems. *Applied Optics*, 34:8409–8421, Dec 1995.

[34] G. Pan, J. Shakal, W. Lai, R. Calabria, P. Massoli, and O. Pust. Simultaneous global size and velocity measurement of droplets and sprays. *Spray '06 – Workshop über Sprays, Erfassung von Sprühvorgängen und Techniken der Fluidzerstäubung*, 2006.

[35] T. Kawaguchi, Y. Akasaka, and M. Maeda. Size measurements of droplets and bubbles by advanced interferometric laser imaging technique. *Meas Sci Technol*, 13:308–316, Feb 2002.

[36] C. Mounaïm-Rousselle and O. Pajot. Droplet sizing by mie scattering interferometry in a spark ignition engine. *Part Part Syst Charact*, 16:160–168, 1999.

[37] N. Damaschke, H. Nobach, and C. Tropea. Optical limits of particle concentration for multi-dimensional particle sizing techniques in fluid mechanics. *Experiments in Fluids*, 32:143–152, 2002.

[38] M. Maeda, Y. Akasaka, and T. Kawaguchi. Improvements of the interferometric technique for simultaneous measurement of droplet size and velocity vector field and its application to a transient spray. *Experiments in Fluids*, 33:125–134, Jun 2002.

[39] N. Damaschke, S. Rasmussen, C. Tropea, and A. Woite. Global phase doppler technique. In *6th International Congress on Optical Particle Characterisation, Brighton, UK*, 2001.

[40] A. Quérel, P. Lemaitre, M. Brunel, E. Porcheron, and G. Gréhan. Real-time processing algorithm of global interferometric laser imaging for the droplet sizing (ilids) in airborne research. In *15th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Jul 2010.

[41] Y. Hardalupas, S. Sahu, A. M. K. P. Taylor, and K. Zarogoulidis. Simultaneous planar measurement of droplet velocity and size with gas phase velocities in a spray by combined ilids and piv techniques. *Experiments in Fluids*, 49:417–434, 2010.

[42] Y. Hardalupas, S. Sahu, and A. M. K. P. Taylor. Measurement of vapour concentration and droplet size and velocity in an evaporative spray. In *15th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Jul 2010.

[43] R. Kurosawa, K. Hishida, and M. Maeda. Combined measurement of lif and ilids for vapor concentration and droplets size and velocity in a spray. In *11th International Symposium on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal*, 2002.

[44] C. Harris and M. Stephens. A combined corner and edge detector. In *Procedings of the Alvey Vision Conference 1988*, pages 23.1–23.6, Jan 1988.

[45] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60, page 91, Nov 2004.

[46] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346–359, 2008.

[47] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011.

[48] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. In *10th European Conference on Computer Vision*, pages 102–115, 2008.

[49] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *2005 IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1508–1515, 2005.

[50] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, Nov 2011.

[51] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *11th European Conference on Computer Vision*, volume LNCS 6314, pages 778–792, 2010.

[52] M. Muja and D. G. Lowe. Fast approximate nearest neighbours with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, pages 331–340, Feb 2009.

[53] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, Jun 1981.

[54] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57:201–218, 2004.

[55] J.-M. Morel and G. Yu. Asift: a new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2:438–469, 2009.

[56] Y.-T. Chi, J. Ho, and M.-H. Yang. A direct method for estimating planar projective transform. In *10th Asian Conference on Computer Vision*, pages 268–281, 2011.

[57] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, Dec 2010.

[58] A. Myronenko and X. Song. Point set registration: coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32:2262–2275, Dec 2010.

[59] C. Doppler. Über das farbige licht der doppelsterne und einiger anderer gestirne des himmels. *Abhandlungen der k. böhm. Gesellschaft der Wissenschaften*, 5:465–482, 1842.

[60] R. C. Whiton, P. L. Smith, S. G. Bigler, K. E. Wilk, and A. C. Harbuck. History of operational use of weather radar by u.s. weather services. part ii: Development of operational doppler weather radars. *Weather and Forecasting*, 13:244–252, 1998.

[61] Y. Yeh and H. Z. Cummins. Localized fluid flow measurements with an he-ne laser spectrometer. *Applied Physics Letters*, 4:176–178, May 1964.

[62] F. Durst and M. Zare. Laser Doppler measurements in two-phase flows. In P. Buchhave, J. M. Delhaye, F. Durst, W. K. George, K. Refslund, and J. H. Whitelaw, editors, *The Accuracy of Flow Measurements by Laser Doppler Methods, Proceedings of the LDA-Symposium '95, Copenhagen, Denmark*, pages 403–429, 1976.

[63] G. Wigley. The sizing of large droplets by laser anemometry. *Journal of Physics E: Scientific Instruments*, 11:639–642, 1978.

[64] K. Thyagarajan and A. Ghatak. *Lasers: Fundamentals and Applications*. Graduate Texts in Physics. Springer, 2010.

[65] M. Saffman. The use of polarized light for optical particle sizing. In *Proceedings of the 3rd International Symposium on Applications of Laser Anemometry to Fluid Mechanics, Lisbon, Portugal*, page 18.2, 1986.

[66] G. Gréhan, G. Gouesbet, and C. Rabasse. The computer program supermidi for lorenz-mie theory and the research of one-to-one relationships for particle sizing. In *Proceedings of the symposium on long range and short range optical velocity measurements*, Institut franco-allemand de Saint-Louis, September 1980.

[67] G. Gouesbet and G. Gréhan. Sur la généralisation de la théorie de lorenz-mie. *Journal of Optics*, 13(2):97–103, 1982.

[68] G. Gouesbet, B. Maheu, and G. Gréhan. Light scattering from a sphere arbitrarily located in a gaussian beam, using a bromwich formulation. *Journal of the Optical Society of America A*, 5(9):1427–1443, 1988.

[69] B. Maheu, G. Gouesbet, and G. Gréhan. A concise presentation of the generalized lorenz-mie theory for arbitrary location of the scatterer in an arbitrary incident profile. *Journal of Optics*, 19(2):59–67, 1988.

[70] J. A. Lock and G. Gouesbet. A rigorous justification of the localized approximation to the beam shape coefficients in generalized lorenz-mie theory, i. on-axis beams. *Journal of the Optical Society of America A*, 11(9):2503–2515, 1994.

[71] G. Gouesbet and J. A. Lock. A rigorous justification of the localized approximation to the beam shape coefficients in generalized lorenz-mie theory, ii. off-axis beams. *Journal of the Optical Society of America A*, 11(9):2516–2525, 1994.

[72] G. Gouesbet and G. Gréhan. Gaussian beam errors in phase-doppler anemometry and their elimination. In R. Adrian, D. Durão, F. Durst, M. Heitor, M. Maeda, and J. Whitelaw, editors, *Developments in Laser Techniques and Applications to Fluid Mechanics*, pages 243–259. Springer Berlin Heidelberg, 1996.

[73] F. Durst, C. Tropea, and T.-H. Xu. The slit effect in phase doppler anemometry. In *Modern techniques and measurements in fluid flows: Proceedings of the 2nd International Conference on Fluid Dynamic Measurement and Its Applications*, Oct 1994.

[74] H. Qiu and C. T. Hsu. Method of phase-doppler anemometry free from the measurement-volume effect. *Applied Optics*, 38(13):2737–2742, May 1999.

[75] C. M. Sipperley and W. D. Bachalo. Triple interval phase doppler interferometry: Improved dense sprays measurements and enhanced phase discrimination. In *17th International Symposium on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal*, Jul 2014.

[76] P. A. Strakey, D. G. Talley, S. V. Sankar, and W. D. Bachalo. Phase Doppler Interferometry with Probe-to-Droplet Size Ratios Less Than Unity. Part I: Trajectory Errors. *Applied Optics*, 39(22):3875–3886, Aug 2000.

[77] P. A. Strakey, D. G. Talley, S. V. Sankar, and W. D. Bachalo. Phase-doppler interferometry with probe-to-droplet size ratios less than unity. part ii. application of the technique. *Appl Opt*, 39:3887–3893, Aug 2000.

[78] J. M. Davis and P. J. Disimile. The effect of optical configuration selection on phase doppler anemometer fire suppressant nozzle characterizations. In *14th Halon Options Technical Working Conference*, May 2004.

Appendix

Detecting overlapping disks using a circular Hough transform

The following Python code uses the circular Hough transform provided by the OpenCV package to detect circular structures in an ILIDS image. The circles are masked by a Gabor patch (i.e. an exponential fade) and a Fourier transform is applied to find a frequency peak within a range of near-horizontal directions. The identified circles, along with the calculated droplet sizes, are shown.

```
import numpy as np
import cv2 as cv
import cv2.cv as cvcv
from skimage.transform import hough_circle
from skimage.feature import peak_local_max
from skimage import filter
from skimage.util import img_as_ubyte

def getDiameter(fringes):
    d_a = 0.021 #m, aperture diameter
    phi = 3.14159/2.0 # 90 degrees, off - axis
    m = 1.333 # relative refractive index
    L = 532e-9 #m, wavelength
    z = 0.45 #m, camera / light sheet distance
    kappa = ((np.arcsin(d_a/(2*z))/L) * (np.cos(phi/2) +
            m * np.sin(phi/2)/np.sqrt(m**2+1-2*m*np.cos(phi/2))))
```

```
    return 1.0 e6 * fringes / kappa

def getCircles(image, radius):
    # use the circular Hough transform to find the circles
    cimg = cv.cvtColor(img, cv.COLOR_GRAY2BGR)
    cimg = cv.GaussianBlur(cimg, (5, 5), 0)
    circles = []
    # apply an edge filter first
    canimg = filter.canny(img_as_ubyte(img), sigma=1, low_threshold
       =120,
                    high_threshold=126)
    hough_radii = np.array([radius])
    hough_res = hough_circle(canimg, hough_radii)
    centers =[]; accums =[]; radii =[]
    numcircles = 9
    for r, h in zip(hough_radii, hough_res):
        peaks = peak_local_max(h, num_peaks = numcircles, min_distance
            =50, threshold_rel=0.15)
        centers.extend(peaks)
        accums.extend(h[peaks[:, 0], peaks[:, 1]])
        radii.extend ([r]* numcircles )
    for i in np.argsort(accums)[::-1][:numcircles]:
        circles.append([centers[i][1], centers[i][0], radii[i]])
    return circles

def getDiameters(image, circles):
    # Apply the Fourier transform to every Gabor - masked circle ,
    # then find the relevant peak frequency and orientation
    h, w = image.shape[:2]
    diameters = []
    for i in circles :
        mask = np.zeros((h, w), np.float)
        cv.circle(mask, (i[0], i[1]), i[2], 255, -1)
        mask = cv.GaussianBlur(mask, (21, 21), 0)
        mask /= 255.0
        maskedimg = mask * image
        f = np.fft.fftshift(np.fft.fft2(maskedimg))
```

```
        f = 20 * np.log(np.abs(f))
        8# find the maximum component
        allowable_freqs = f[(h/2)-(h/7):(h/2)+(h/7), (w/2)+10:]
        maxfreq_x_y = np.unravel_index(allowable_freqs.argmax () ,
                allowable_freqs.shape)
        # x - fringes / imagewidth : maxfreq_x_y[1]+19
        xfringes =(maxfreq_x_y[1]+10)* i[2]*2/ w
        yfringes = abs(maxfreq_x_y[0])* i[2]*2/ h
        totalfringes = np.sqrt(xfringes ** 2 + yfringes ** 2)
        print(xfringes, yfringes, totalfringes)
        # x - fringes / dropsize :(maxfreq_x_y[ 1 ] + 1 9 ) * i[2]*2/
           w
        diameters.append(str(int(totalfringes)) +"("+
                        str(int(getDiameter(totalfringes ))) +"um )")
    return diameters

def drawDropletSizes(image, circles, diameters):
h, w = image.shape[:2]
circles = np.uint16(np.around(circles))
img = cv.cvtColor(image, cv.COLOR_GRAY2BGR)
for counter, i in enumerate(circles):
    cv.circle(img ,(i[0], i[1]), i[2] + 10, (0, 255, 0), 1)
    cv.putText(img, diameters[counter] ,(i[0] - 20, i[1]) ,
    cv.FONT_HERSHEY_SIMPLEX, 0.3, (0, 0, 255))
cv.imshow("detected circles", img)
cv.waitKey(0)
cv.destroyAllWindows()
img = cv.imread("image_cropped.jpg", 0)
fullimg = cv.imread("image_cropped.tif", -1)
circles = getCircles(img, 42)
diameters = getDiameters(fullimg, circles)
drawDropletSizes(img, circles, diameters)
```

Detection of stripes and ILIDS size analysis

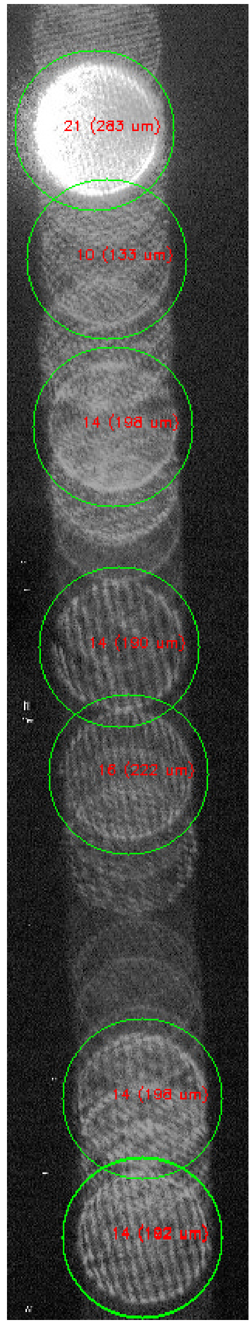


Figure A.1: Droplet sizes as estimated by the above script. Not that only some of the droplets were identified correctly by the circular Hough algorithm.

```
import numpy as np
import scipy.fftpack as fftpack
from skimage import filter
from skimage import io
from skimage.feature import match_template

# first, import the image
img = io.imread("image.tif", True)

# create rectangle
stripheight = 14
stripwidth = 120

pattern = 255 * np.ones((stripheight, stripwidth))

# hanning-window it
pattern *= np.hanning(stripheight)[:np.newaxis]
pattern *= np.hanning(stripwidth)[:np.newaxis].T

# correlate
cor = match_template(img, pattern)

def find_peaks(correlation_result):
    # 1. identify peak
    # 2. remove from image a rectangle until no more peaks are above a
    # threshold

    xs = []
    ys = []
    threshold = 200
    while(np.max(correlation_result) > threshold):
        x = np.argmax(correlation_result, axis=1)
        y = np.argmax(correlation_result, axis=0)
        correlation_result[y-stripheight/3:y+stripheight/3,
                x-stripwidth/3:x+stripwidth/3] = 0
        xs.append(x)
        ys.append(y)
```

```
    return (xs, ys)

# find peaks
xs, ys = find_peaks(cor)

def find_fringecount(stripimg):
    # apply hanning
    stripimg *= np.hanning(stripheight)[:np.newaxis]
    stripimg *= np.hanning(stripwidth)[:np.newaxis].T
    # pad it
    padded = np.zeros((1024,1024))
    padded[512-stripimg.shape[0]/2:512+stripimg.shape[0]/2,
           512-stripimg.shape[1]/2:512+stripimg.shape[1]/2] = stripimg
    # FFT it
    fftimg = fftpack.fftshift(fftpack.fft2(padded))

    fftheight = 40
    fftwidth = 400
    # take center
    fftimg = fftimg[512-fftheight/2:512+fftheight/2,
                    512, 512+fftwidth]
    # take mean and clip
    fftimg = np.mean(fftimg, axis=0)
    fftimg[:,0:40] = 0
    fftimg[:,150:] = 0
    # find max and get fringe count
    return stripwidth / (1024/np.argmax(fftimg))

fringecounts = []

# for every peak, take rectangle
for (x, y) in zip(xs, ys):
    # deal with edge cases later
    if (y < stripheight/2 or y > img.shape[0]-stripheight/2
        or x < stripwidth/2 or x > img.shape[1]-stripwidth/2):
        continue
    strip = img[y-stripheight/2:y+stripheight/2,
```

```
                x-stripwidth/2:x+stripwidth/2]
    fringecounts.append(find_fringecount(strip))

# plot various things
# (not shown here)

print(fringecounts)
```

Homography estimation using ASIFT

```
import numpy as np
import cv2
import scipy.ndimage.morphology.grey_erosion
from skimage.draw import circle
from scipy.misc import imresize

# From the OpenCV examples:
# built-in modules
import itertools as it
from multiprocessing.pool import ThreadPool

# local modules
from common import Timer
from find_obj import init_feature, filter_matches, explore_match

# From OpenCV samples
def affine_skew(tilt, phi, img, mask=None):
    '''
    affine_skew(tilt, phi, img, mask=None) -> skew_img, skew_mask, Ai

    Ai - is an affine transform matrix from skew_img to img
    '''
    h, w = img.shape[:2]
    if mask is None:
        mask = np.zeros((h, w), np.uint8)
        mask[:] = 255
```

```
    A = np.float32([[1, 0, 0], [0, 1, 0]])
    if phi != 0.0:
        phi = np.deg2rad(phi)
        s, c = np.sin(phi), np.cos(phi)
        A = np.float32([[c,-s], [ s, c]])
        corners = [[0, 0], [w, 0], [w, h], [0, h]]
        tcorners = np.int32( np.dot(corners, A.T) )
        x, y, w, h = cv2.boundingRect(tcorners.reshape(1,-1,2))
        A = np.hstack([A, [[-x], [-y]]])
        img = cv2.warpAffine(img, A, (w, h), flags=cv2.INTER_LINEAR,
            borderMode=cv2.BORDER_REPLICATE)
    if tilt != 1.0:
        s = 0.8*np.sqrt(tilt*tilt-1)
        img = cv2.GaussianBlur(img, (0, 0), sigmaX=s, sigmaY=0.01)
        img = cv2.resize(img, (0, 0), fx=1.0/tilt, fy=1.0,
            interpolation=cv2.INTER_NEAREST)
        A[0] /= tilt
    if phi != 0.0 or tilt != 1.0:
        h, w = img.shape[:2]
        mask = cv2.warpAffine(mask, A, (w, h), flags=cv2.INTER_NEAREST
            )
    Ai = cv2.invertAffineTransform(A)
    return img, mask, Ai


def affine_detect(detector, img, mask=None, pool=None):
    '''
    affine_detect(detector, img, mask=None, pool=None) -> keypoints,
        descrs

    Apply a set of affine transormations to the image, detect
        keypoints and
    reproject them into initial image coordinates.
    See http://www.ipol.im/pub/algo/my_affine_sift/ for the details.

    ThreadPool object may be passed to speedup the computation.
    '''
```

```
    params = [(1.0, 0.0)]
    for t in 2**(0.5*np.arange(1,6)):
        for phi in np.arange(0, 180, 72.0 / t):
            params.append((t, phi))

    def f(p):
        t, phi = p
        timg, tmask, Ai = affine_skew(t, phi, img)
        keypoints, descrs = detector.detectAndCompute(timg, tmask)
        for kp in keypoints:
            x, y = kp.pt
            kp.pt = tuple( np.dot(Ai, (x, y, 1)) )
        if descrs is None:
            descrs = []
        return keypoints, descrs

    keypoints, descrs = [], []
    if pool is None:
        ires = it.imap(f, params)
    else:
        ires = pool.imap(f, params)

    for i, (k, d) in enumerate(ires):
        print 'affine sampling: %d / %d\r' % (i+1, len(params)),
        keypoints.extend(k)
        descrs.extend(d)

    print
    return keypoints, np.array(descrs)



img1 = cv2.imread("masked_points.jpg", 0)
img2 = cv2.imread("masked_disks.jpg", 0)

# pixel-wise mean taken over an image ensemble
mean1 = cv2.imread("masked_mean1.jpg", 0)
```

```
mean2 = cv2.imread("masked_mean2.jpg", 0)

# turn the points into disks
img1 -= mean1
img2 -= mean2

img1 = grey_erosion(img1, size=(3, 3))

# flip and scale the image
img1 = np.hflip(img1)
img1 = imresize(0.5)

# locate intensity peaks. wherever they are, black them out and
    replace them
with a large circle in a blank image

virtualdisks = np.zeros(img2.shape)
max_threshold = 200
virtualradius = 60
eraserradius = 13
while np.max(img1) > 200:
    cy, cx = np.argmax(img1, axis=0), np.argmax(img1, axis=1)
    rr, cc = circle(cy, cx, virtualradius)
    virtualdisks[rr, cc] = 255
    re, ce = circle(cy, cx, eraserradius)
    img1[re, ce] = 0

# now match using asift (from OpenCV samples)

detector, matcher = init_feature('orb')
pool=ThreadPool(processes = cv2.getNumberOfCPUs())
kp1, desc1 = affine_detect(detector, virtualdisks, pool=pool)
kp2, desc2 = affine_detect(detector, img2, pool=pool)
def match_and_draw(win):
    with Timer('matching'):
        raw_matches = matcher.knnMatch(desc1, trainDescriptors = desc2
            , k = 2) #2
```

```
    p1, p2, kp_pairs = filter_matches(kp1, kp2, raw_matches)
    if len(p1) >= 4:
        H, status = cv2.findHomography(p1, p2, cv2.RANSAC, 5.0)
        print '%d / %d  inliers/matched' % (np.sum(status), len(status
            ))
        # do not draw outliers (there will be a lot of them)
        kp_pairs = [kpp for kpp, flag in zip(kp_pairs, status) if flag
            ]
    else:
        H, status = None, None
        print '%d matches found, not enough for homography estimation'
             % len(p1)

    vis = explore_match(win, img1, img2, kp_pairs, None, H)

# show the match
match_and_draw('affine find_obj')
cv2.waitKey()
cv2.destroyAllWindows()

# compute the homography
doubleSize = np.array([[2, 0, 0], [0, 2, 0], [0, 0, 1]])
flipV = np.array([[1, 0, 0], [0, -1, img1.shape[0]], [0, 0, 1]])
Hh = np.dot(np.dot(H, doubleSize), flipV)

Pfoc = np.array() # from DantecStudio, or some constant
Pdef = np.dot(Hh, Pfoc)
print(Pfoc)
print(Pdef)
```

The above script relies on `find_obj.py`, which is part of the OpenCV samples collection and reproduced below.

```
#!/usr/bin/env python

'''
OpenCV sample file
```

```
Feature-based image matching sample.

Note, that you will need the https://github.com/Itseez/opencv_contrib
    repo for SIFT and SURF

USAGE
  find_obj.py [--feature=<sift|surf|orb|akaze|brisk>[-flann]] [ <
      image1> <image2> ]

  --feature  - Feature to use. Can be sift, surf, orb or brisk. Append
       '-flann'
               to feature name to use Flann-based matcher instead
                   bruteforce.

  Press left mouse button on a feature point to see its matching point
      .
'''

import numpy as np
import cv2
from common import anorm, getsize

FLANN_INDEX_KDTREE = 1  # bug: flann enums are missing
FLANN_INDEX_LSH    = 6


def init_feature(name):
    chunks = name.split('-')
    if chunks[0] == 'sift':
        detector = cv2.xfeatures2d.SIFT_create()
        norm = cv2.NORM_L2
    elif chunks[0] == 'surf':
        detector = cv2.xfeatures2d.SURF_create(800)
        norm = cv2.NORM_L2
    elif chunks[0] == 'orb':
        detector = cv2.ORB_create(400)
        norm = cv2.NORM_HAMMING
```

```
    elif chunks[0] == 'akaze':
        detector = cv2.AKAZE_create()
        norm = cv2.NORM_HAMMING
    elif chunks[0] == 'brisk':
        detector = cv2.BRISK_create()
        norm = cv2.NORM_HAMMING
    else:
        return None, None
    if 'flann' in chunks:
        if norm == cv2.NORM_L2:
            flann_params = dict(algorithm = FLANN_INDEX_KDTREE, trees
                = 5)
        else:
            flann_params= dict(algorithm = FLANN_INDEX_LSH,
                               table_number = 6, # 12
                               key_size = 12,     # 20
                               multi_probe_level = 1) #2
        matcher = cv2.FlannBasedMatcher(flann_params, {})  # bug :
            need to pass empty dict (#1329)
    else:
        matcher = cv2.BFMatcher(norm)
    return detector, matcher


def filter_matches(kp1, kp2, matches, ratio = 0.75):
    mkp1, mkp2 = [], []
    for m in matches:
        if len(m) == 2 and m[0].distance < m[1].distance * ratio:
            m = m[0]
            mkp1.append( kp1[m.queryIdx] )
            mkp2.append( kp2[m.trainIdx] )
    p1 = np.float32([kp.pt for kp in mkp1])
    p2 = np.float32([kp.pt for kp in mkp2])
    kp_pairs = zip(mkp1, mkp2)
    return p1, p2, kp_pairs

def explore_match(win, img1, img2, kp_pairs, status = None, H = None):
```

```
h1, w1 = img1.shape[:2]
h2, w2 = img2.shape[:2]
vis = np.zeros((max(h1, h2), w1+w2), np.uint8)
vis[:h1, :w1] = img1
vis[:h2, w1:w1+w2] = img2
vis = cv2.cvtColor(vis, cv2.COLOR_GRAY2BGR)

if H is not None:
    corners = np.float32([[0, 0], [w1, 0], [w1, h1], [0, h1]])
    corners = np.int32( cv2.perspectiveTransform(corners.reshape
        (1, -1, 2), H).reshape(-1, 2) + (w1, 0) )
    cv2.polylines(vis, [corners], True, (255, 255, 255))

if status is None:
    status = np.ones(len(kp_pairs), np.bool_)
p1 = np.int32([kpp[0].pt for kpp in kp_pairs])
p2 = np.int32([kpp[1].pt for kpp in kp_pairs]) + (w1, 0)

green = (0, 255, 0)
red = (0, 0, 255)
white = (255, 255, 255)
kp_color = (51, 103, 236)
for (x1, y1), (x2, y2), inlier in zip(p1, p2, status):
    if inlier:
        col = green
        cv2.circle(vis, (x1, y1), 2, col, -1)
        cv2.circle(vis, (x2, y2), 2, col, -1)
    else:
        col = red
        r = 2
        thickness = 3
        cv2.line(vis, (x1-r, y1-r), (x1+r, y1+r), col, thickness)
        cv2.line(vis, (x1-r, y1+r), (x1+r, y1-r), col, thickness)
        cv2.line(vis, (x2-r, y2-r), (x2+r, y2+r), col, thickness)
        cv2.line(vis, (x2-r, y2+r), (x2+r, y2-r), col, thickness)
vis0 = vis.copy()
for (x1, y1), (x2, y2), inlier in zip(p1, p2, status):
```

```
        if inlier:
            cv2.line(vis, (x1, y1), (x2, y2), green)

    cv2.imshow(win, vis)
    def onmouse(event, x, y, flags, param):
        cur_vis = vis
        if flags & cv2.EVENT_FLAG_LBUTTON:
            cur_vis = vis0.copy()
            r = 8
            m = (anorm(p1 - (x, y)) < r) | (anorm(p2 - (x, y)) < r)
            idxs = np.where(m)[0]
            kp1s, kp2s = [], []
            for i in idxs:
                 (x1, y1), (x2, y2) = p1[i], p2[i]
                 col = (red, green)[status[i]]
                 cv2.line(cur_vis, (x1, y1), (x2, y2), col)
                 kp1, kp2 = kp_pairs[i]
                 kp1s.append(kp1)
                 kp2s.append(kp2)
            cur_vis = cv2.drawKeypoints(cur_vis, kp1s, flags=4, color=
                kp_color)
            cur_vis[:,w1:] = cv2.drawKeypoints(cur_vis[:,w1:], kp2s,
                flags=4, color=kp_color)

        cv2.imshow(win, cur_vis)
    cv2.setMouseCallback(win, onmouse)
    return vis


if __name__ == '__main__':
    print __doc__

    import sys, getopt
    opts, args = getopt.getopt(sys.argv[1:], '', ['feature='])
    opts = dict(opts)
    feature_name = opts.get('--feature', 'sift')
    try:
```

```
    fn1, fn2 = args
except:
    fn1 = '../data/box.png'
    fn2 = '../data/box_in_scene.png'

img1 = cv2.imread(fn1, 0)
img2 = cv2.imread(fn2, 0)
detector, matcher = init_feature(feature_name)

if img1 is None:
    print 'Failed to load fn1:', fn1
    sys.exit(1)

if img2 is None:
    print 'Failed to load fn2:', fn2
    sys.exit(1)

if detector is None:
    print 'unknown feature:', feature_name
    sys.exit(1)

print 'using', feature_name

kp1, desc1 = detector.detectAndCompute(img1, None)
kp2, desc2 = detector.detectAndCompute(img2, None)
print 'img1 - %d features, img2 - %d features' % (len(kp1), len(
    kp2))

def match_and_draw(win):
    print 'matching...'
    raw_matches = matcher.knnMatch(desc1, trainDescriptors = desc2
        , k = 2) #2
    p1, p2, kp_pairs = filter_matches(kp1, kp2, raw_matches)
    if len(p1) >= 4:
        H, status = cv2.findHomography(p1, p2, cv2.RANSAC, 5.0)
        print '%d / %d  inliers/matched' % (np.sum(status), len(
            status))
```

```
    else:
        H, status = None, None
        print '%d matches found, not enough for homography
            estimation' % len(p1)

    vis = explore_match(win, img1, img2, kp_pairs, status, H)

match_and_draw('find_obj')
cv2.waitKey()
cv2.destroyAllWindows()
```

Sample set of raw PDPA data

The following data points were acquired for a combination of 110.7 ml/h and 1522 Hz. Only the first 50 rows are shown.

Diameter (um)	Intensity (mV)	AB Phase (deg)	AC Phase (deg)
114.719231	35.15625	51.328125	153.896484
163.368668	75.613838	71.367188	-144.580078
106.222809	23.716518	40.517578	150.380859
114.598114	65.011162	49.306641	155.742188
119.298218	63.058037	45	167.871094
101.284721	83.426338	44.033203	138.603516
129.449677	56.919643	49.658203	-179.736328
106.529572	78.683037	46.054688	145.458984
103.525955	69.475449	47.460938	138.955078
307.841827	183.872772	129.902344	39.550781
117.489494	81.473213	60.380859	149.501953
120.334213	140.345978	45.703125	168.925781
100.29361	65.569199	49.658203	131.308594
115.773041	65.290176	50.273438	156.708984
114.002083	54.6875	45.263672	158.730469
105.346077	42.410713	38.583984	150.908203
114.519005	40.736607	46.318359	158.554688
110.710243	28.180803	46.582031	151.875

222.697067	96.261162	91.142578	-64.863281
99.281479	61.38393	32.607422	146.689453
69.957497	16.462053	29.707031	100.371094
254.884995	158.482147	99.492188	-19.248047
132.176956	99.609375	62.666016	171.826172
109.313354	109.375	47.548828	148.535156
101.030701	46.595982	36.386719	145.810547
99.588203	20.647322	44.384766	135.351563
142.063126	119.140625	74.003906	177.099609
113.804352	30.970982	50.097656	153.544922
158.861282	125.279015	70.400391	-151.171875
142.957535	131.417404	58.798828	-166.289063
139.260284	105.747765	61.962891	-175.693359
119.566116	34.04018	54.755859	158.554688
121.7621	67.522324	59.589844	157.412109
95.901268	20.647322	56.337891	117.158203
128.383881	118.303574	56.865234	171.298828
108.952812	32.645088	55.810547	139.658203
115.696075	107.14286	52.646484	154.160156
139.023727	132.254471	58.359375	-172.441406
188.533722	91.796875	62.578125	-93.427734
157.50322	79.241074	70.576172	-153.632813
104.477509	28.459822	39.814453	148.183594
114.200165	61.104912	46.142578	158.203125
213.802048	187.220978	90.351563	-78.837891
124.05854	119.41964	60.46875	160.400391
106.191475	29.017857	42.890625	148.007813
125.415169	62.220982	54.140625	169.013672
98.739624	51.89732	38.759766	139.658203
163.894836	66.127235	61.171875	-133.59375
124.03717	70.591515	71.367188	149.501953
125.491409	63.61607	53.173828	170.068359