Resolving Gas Bubbles Ascending in Liquid Metal from Low-SNR Neutron Radiography Images

Mihails Birjukovs [1,*], Pavel Trtik [2,*], Anders Kaestner [2], Jan Hovind [2], Martins Klevs [1], Dariusz Jakub Gawryluk [3], Knud Thomsen [2] and Andris Jakovics [1]

[1] Institute of Numerical Modelling, University of Latvia, Jelgavas 3, 1004 Riga, Latvia; martinsklevs@gmail.com (M.K.); andris.jakovics@lu.lv (A.J.)

[2] Paul Scherrer Institut, Research with Neutrons and Muons, Forschungsstrasse 111, 5232 Villigen, Switzerland; anders.kaestner@psi.ch (A.K.); jan.hovind@psi.ch (J.H.); knud.thomsen@psi.ch (K.T.)

[3] Paul Scherrer Institut, Laboratory for Multiscale Materials Experiments, Forschungsstrasse 111, 5232 Villigen, Switzerland; dariusz.gawryluk@psi.ch

[*] Correspondence: mihails.birjukovs@lu.lv (M.B.); pavel.trtik@psi.ch (P.T.)

Abstract: We demonstrate a new image processing methodology for resolving gas bubbles travelling through liquid metal from dynamic neutron radiography images with an intrinsically low signal-to-noise ratio. Image pre-processing, denoising and bubble segmentation are described in detail, with practical recommendations. Experimental validation is presented—stationary and moving reference bodies with neutron-transparent cavities are radiographed with imaging conditions representative of the cases with bubbles in liquid metal. The new methods are applied to our experimental data from previous and recent imaging campaigns, and the performance of the methods proposed in this paper is compared against our previously achieved results. Significant improvements are observed as well as the capacity to reliably extract physically meaningful information from measurements performed under highly adverse imaging conditions. The showcased image processing solution and separate elements thereof are readily extendable beyond the present application, and have been made open-source.

Keywords: dynamic neutron imaging; image processing; denoising; segmentation; low signal-to-noise ratio (SNR); bubble flow; liquid metal; two-phase flow; metohydrodynamics (MHD)

1. Introduction

Gas bubble flow in liquid metal is encountered in a variety of industrial processes. Examples include liquid metal stirring, purification and continuous casting in metallurgy, liquid metal-based chemical reactors, and more. Controlling the output of these processes is essential and it was proposed that this can be (and in some cases already is) achieved via applied magnetic field (MF) [1–8].

Single-bubble flow with and without applied MF has been systematically studied using ultrasound Doppler velocimetry (UDV) [9–12], ultrasound transit time techniques [13,14], X-ray imaging [14] and numerical simulations [15–22]. The most important features and physical mechanisms involved in single-bubble flow are presently rather well known [22–27], but many aspects of bubble collective dynamics, especially in the presence of MF, are not fully understood or have not been studied in-depth [28–31]. Clearly, this is a problem from the process engineering and optimization perspective, and also from the point of view of computational fluid dynamics (CFD) where it is of interest to improve effective models for bubble flow (Euler–Euler and Lagrangian) [2,32–35].

Through recent efforts and the advent of dynamic X-ray and neutron radiography of two-phase liquid metal flow [36–42], fundamental investigation of bubble chain systems mimicking industrially relevant flow conditions is underway [1,8,15,28,29,36,43–46]. In bubble chain flow, bubbles are released into a liquid metal system one-by-one with a

uniform time delay between each, at a certain gas flow rate, and ascend to the free surface of liquid metal. Such systems are usually rectangular vessels filled with gallium [1,8,43] or an eutectic gallium–indium–tin alloy [28,29,36,44,46] where bubbles are introduced via horizontal [8,43] or vertical [1,28,29,44,46] nozzles at the bottom of the vessel, or top-submerged vertical [36] nozzles. Bubble chain flow systems are the next logical step from single-bubble flow investigations, since single-bubble flow, while very informative of the bubble wake flow dynamics and characteristic trajectories without and with applied MF, is not representative of the actual flow conditions typical for the above mentioned industrial processes where one has columns and swarms with a high number density of deformable bubbles [47–50].

Bubble chains are still simple enough to enable experimentation with compact systems [1,8,43,45,46] and contain computationally manageable numbers of bubbles within the liquid metal volume [8,15,43,45]. Meanwhile, they already exhibit collective dynamics between leading and trailing bubbles [8,15,43–45] and, depending on the system geometry and flow rate, bubble agglomeration, coalescence and breakup can occur [28,29,36].

Therefore these systems are a crucial milestone in a transition from studying single-bubble flow to investigations of many-bubble systems that are very close to their actual industrial counterparts. However, despite the *relative simplicity*, dynamics exhibited by bubble chain flow in liquid metal without or with applied MF are still very complex. Depending on the gas flow rate, bubbles produce unstable elongated wake flow regions where periodic vortex detachment occurs and turbulent pulsations are generated—shed vortices and turbulent wakes of leading bubbles strongly affect the trailing bubbles, leading to bubble pair coupling across the ascending chain [8,15,43,45,51–54]. There exists a feedback loop involving combined perturbations of bubble shapes and within the bubble chain, surrounding liquid metal flow, and the influence of the free surface at the top of the metal vessels with instabilities and oscillations in the bubble chain shape [8,15,43,45,54]. Recently, dynamic mode decomposition (DMD) has been applied to the output of the MHD bubble chain flow simulation to study both large-scale flow structures and bubble wake flow in the bubble reference frame [55]. It has been demonstrated that DMD is a viable tool for an in-depth analysis of the complex dynamics mentioned above, and it was shown that there exists a very complex interplay of bubble wake flow and large-scale flow modes with a wide range of spatial and temporal scales.

It has become clear that specialized and rather advanced image processing methods and tools are required to extract physically meaningful data from data sets acquired via dynamic neutron and/or X-ray imaging [1,8,43]. This is mainly due to the low signal-to-noise ratio (SNR) associated with imaging thick (>20–30 mm) layers of liquid metal at frame rates $\gtrsim$100 frames per second (FPS) and the need to resolve many often closely packed interacting objects. High frame rates are a requirement to enable capturing fast bubbles, drops and particles flowing in liquid metal and to avoid motion blur [1,8,43]. Meanwhile, neutron flux that can be used in experiments is limited by both the utilized neutron source and the rapid activation of model liquid metals such as gallium.

Here we describe and demonstrate our new image processing methodology developed over the course of our dynamic neutron imaging experiments with bubble flow in liquid metal. We show that the implemented code is robust and can operate reliably at very low SNR in the presence of image artefacts. It is also shown that this version clearly outperforms our previous solution by resolving/avoiding known issues. In addition to demonstrating performance for dynamic neutron imaging data sets from our measurements with a rectangular gallium vessel with bubble chain flow, we have also performed direct experimental validation of the code by imaging a reference spherical body, both stationary and in motion, and have quantified the shape detection errors.

2. Image and Bubble Characterization

2.1. Image Acquisition

Bubble flow imaging with and without applied magnetic field was conducted at the thermal neutron imaging beamline NEUTRA (SINQ, PSI, 20 mm aperture, 10^7 n cm^{-2} s^{-1} mA^{-1} flux) at the Paul Scherrer Institute (PSI) using a medium spatial resolution set-up (MIDI) [56]. Experiments were performed with both horizontally and vertically oriented inlet nozzles. The setup with the horizontal inlet is described in [8,43], whereas a modified version of that model gallium/argon system was designed for the new experiments with the vertical inlet—the latter will be described in a follow-up publication. A thin-walled (4 mm) rectangular 150 mm × 90 mm (95 mm) × 30 mm (interior dimensions) glass (boron-free) vessel filled with liquid gallium (linear attenuation coefficient, LAC, for neutrons at NEUTRA is 0.56 cm^{-1}) up to the 130- to 140-mm mark was used. Neutron flux was parallel to the 30-mm dimension of the vessel. A square field of view (FOV, 112.8- or 123.125-mm side) above the gas inlet was imaged at 100 FPS. The distance between the liquid metal layer and the scintillator and was [4; 32] mm depending on the setup (magnetic field system used, if any). A sCMOS *ORCA Flash 4.0* camera was used to collect the scintillator (200 μm thick $^6LiF/ZnS$ screen).

Reference experiments were performed at the cold neutron beamline ICON (SINQ, PSI, 20 mm aperture, ∼1.3 × NEUTRA flux) to validate the developed image processing methodology [57]. A 20 mm × 20 mm × 30 mm rectangular brass reference body (stationary and moving, LAC at ICON is 0.90 cm^{-1}) with a central spherical cavity (5 mm radius) was imaged at 100 FPS within a 120.06 mm square FOV (a sCMOS *ORCA Flash 4.0 V2* camera with a 200 μm $^6LiF/ZnS$ scintillator) to reproduce the imaging conditions for argon bubbles in liquid gallium. The reference body was imaged with neutron flux directed along its shorter or longer axes. In addition, the distance between the scintillator and the body was either 0- (static body)/2 mm (moving body) or with an extra 1 cm. These adjustments to the test conditions were meant to determine how the reference shape acquisition error depends on SNR which these conditions modify.

For each image sequence recording at the NEUTRA and ICON camera, dark current signals and neutron beam profile signals were recorded to be used for subsequent image normalization during pre-processing.

2.2. Image Properties

The acquired images are 16-bit 1-channel TIFFs with a 1024 × 1024 pixel resolution (2 × 2 average-binned 2048 × 2048 frames) with a 0.11–0.12 mm pixel size (0.12 mm for the reference experiments). Short exposure times (10 ms) result in strong Poisson (multiplicative) noise from neutrons and converted photons, and salt-and-pepper noise of varying density is present due to overexposed (gamma ray noise) or "dead" camera pixels. The neutron beam flux over the FOV is non-uniform with a fall-off near the edges of the acquired images.

Figure 1a is an example of an acquired raw image. Here gas flow rate was 120 sccm (standard cubic centimeters per minute) and static vertical magnetic field of 125 mT was applied to the bubble chain region. Typically one can see the following captured in a full FOV image: the liquid gallium volume within the glass container (delimited by the interior orange lines and the light blue line), the container walls (orange lines), the free surface of liquid metal (the light blue line), the surrounding air (regions outside the walls and above the metal free surface) and the neutron flux shielding (borated polyethylene, the red line) for the magnetic field system. However, not all of the FOV is of interest—Figure 1b shows the FOV cropped to the liquid metal volume in false color after pre-processing (Section 3.2). Note also the bubble regions highlighted in Figure 1a,b—these are the objects of interest that must be segmented and their properties such as centroids, projection areas, tilt angles, aspect ratios, etc. measured. Note that the same color scheme and normalized luminance scale as in Figure 1b are used in all further figures unless stated otherwise.
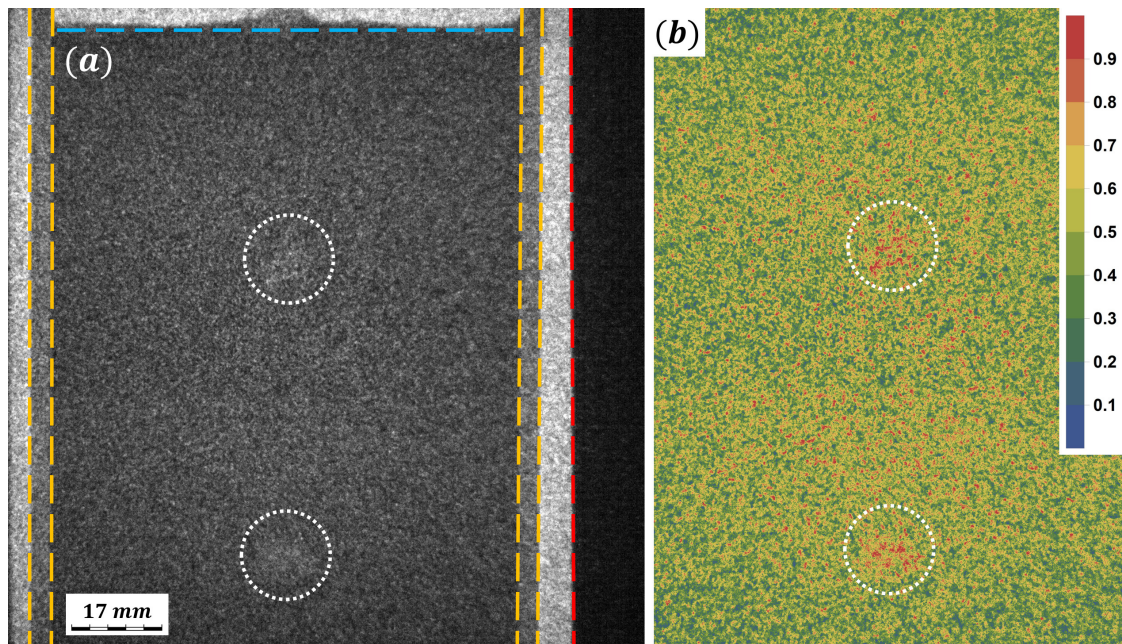
Figure 1. (**a**) Original captured FOV after outlier removal and luminance normalization with marked container walls (orange dashed lines), the mean gallium free surface level (light blue), the neutron flux shielding (red, borated polyethylene) and bubble locations within the FOV (white). Note the scale bar in the bottom-left corner. (**b**) FOV in false color (color bar on the right) after cropping to the container walls and the metal free surface, dark current and flat field corrections, and normalization.

Figures 2 and 3 illustrate the image processing challenge with examples of image noise and neutron flux transmission signal for bubbles after pre-processing (Section 3.2). Figure 2 shows a typical luminance distribution about a bubble region: here, moving averages (window width in the scan direction equal to one pixel) over (a) horizontal and (b) vertical image patches are shown, since plots over any given single pixel line would be unintelligible. Estimates from a control set of bubble regions for different frames indicate many of the images have SNR $= \langle S_{\mathrm{b}}^2 \rangle / \sigma_{\mathrm{n}}^2 \sim 1.5$–1.6 ($S_{\mathrm{b}}$ is the bubble signal intensity, σ_{n} is the noise standard deviation), and there are cases with SNR as low as $\sim$1.2. Similar plots for (a) horizontal and (b) vertical patches for the entire width/length of the cropped FOV are shown in Figure 3.
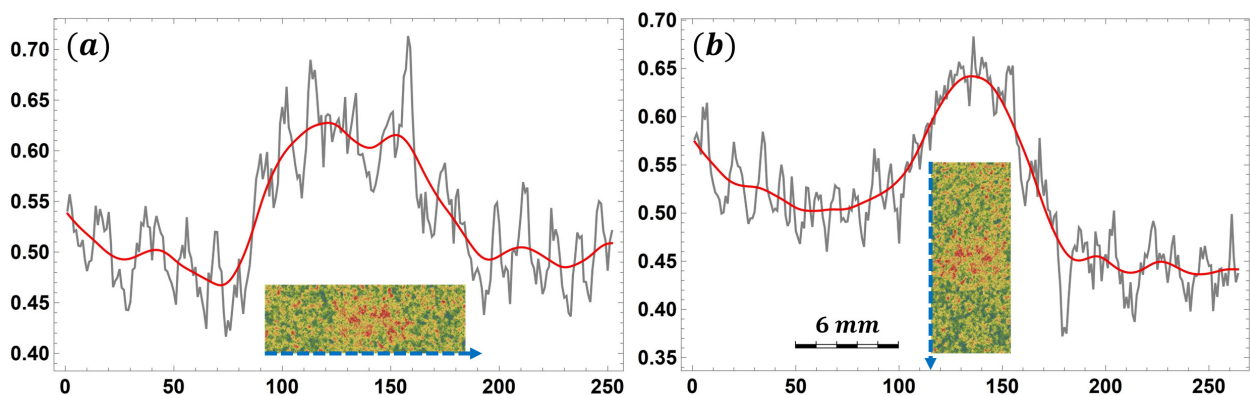

Figure 2. Bubble neighborhood analysis: width mean of luminance (gray) over the length of (**a**) horizontal and (**b**) vertical patches roughly fitted to the bubble region within the FOV. Normalized luminance versus pixel coordinates is shown in both cases. Note the pixel-to-*mm* scale bar in (**b**). Bubble neighborhood patches shown in (**a**,**b**) are not to scale and are normalized as in Figure 1. Scan directions are indicated by the dashed blue arrows and the red curve is the total variation filtered (Gaussian, regularization parameter equal to 1 [58]) gray curve.

While visible in (a), it is especially evident in (b) that despite the compensation for neutron flux non-uniformity over the FOV, the background "mean" is still not uniform/flat. One also has rather densely packed noise peaks with luminance values comparable to the values associated with the bubbles.
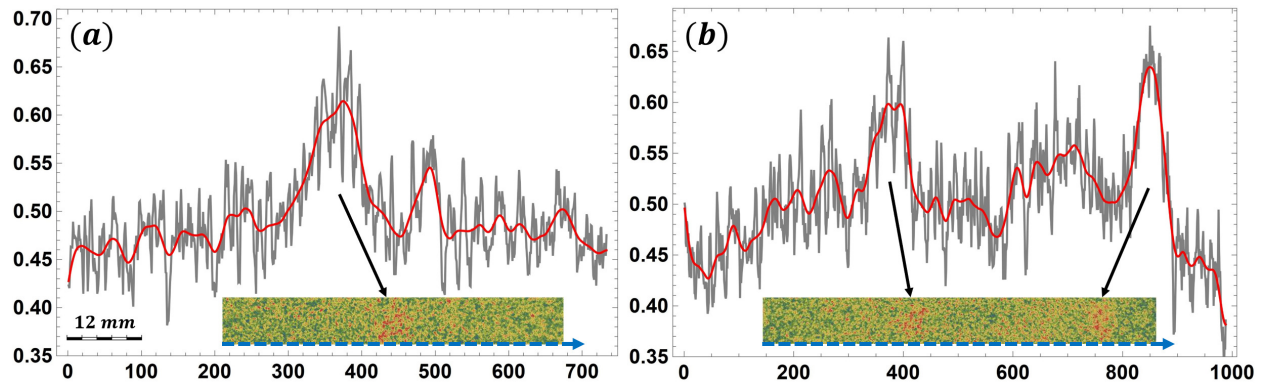

Figure 3. Bubble signal analysis for the cropped FOV: width mean of luminance (gray) over the length of (**a**) horizontal and (**b**) vertical patches roughly fitted to the bubble dimensions, spanning the FOV. Note the scale bar in (**a**).

2.3. Bubble Flow Properties

Estimates indicate that Eötvös number is $Eo \in [2.1; 4.1]$ and the Reynolds number near the bubbles is within $Re \in [10^3; 10^4]$. This corresponds to a flow regime wherein bubbles have oscillating elliptic shapes [8,24,43,59]. Bubbles have equivalent diameters of $d_b \in [6; 8]$ mm and travel with varying velocities usually in the [20; 40] cm/s range [43]. Such velocities and dynamic phase boundaries dictate the high acquisition frame rate required for physical analysis. A frame rate of 100 FPS was considered an optimal trade-off for the experiments covered herein, in that one already avoids significant motion blur about the bubbles while still maintaining manageable image SNR.

Bubbles ascend in vertical chains from the bottom of the gallium vessel to the free surface where they exit the liquid metal volume. Bubbles generally exhibit non-uniformly accelerated motion depending on the applied magnetic field and flow rate. Bubble trajectories are mostly planar zigzags (plane parallel to the largest container face) with slight out-of-plane perturbations that intensify with gas flow rate. Bubble collisions, coalescence or breakup do not occur for image sequences considered in this paper, but rather chains with closely packed bubbles manifest for higher flow rates. In the future, the methodology developed herein will also be applied for even higher flow rate cases where bubble collisions do take place. Here we focus on capturing the bubble shape variations and positions.

3. Image Processing

3.1. Assumptions and Considerations for Algorithm Development

The results obtained via the previous version of our image processing code [8,43] determined the objectives that must be met here:

1. Improved bubble edge detection stability—more reliable detection, less artefacts and false positives;
2. Greater bubble shape detection precision;
3. Increased bubble detection rates at the bottom of the FOV.

The following assumptions regarding the bubbles are in effect:

- Bubbles have perturbed elliptic/circular shapes, i.e., not necessarily convex.
- Bubble phase boundaries do not exhibit local curvature radii less that a pre-defined fraction of their mean curvature, i.e., smoothness is assumed below a certain length scale due to surface tension.

- No coalescence or breakup of bubbles is expected at flow rates <500 sccm considered herein.

In addition, the development of the methodology outlined herein was subject to the following considerations:

- Given the image properties/quality as described in Section 2.2, we do not attempt to perform filtering and segmentation such that argon/gallium volume fraction can be recovered, since low SNR will translate to significant errors in the output if both volume fraction field and shape recovery are attempted.
- Since the 100 FPS frame rate is the lower limit for prevention of considerable motion blur, we do not employ methods that perform spatio-temporal noise filtering—only spatial denoising is used.
- Furthermore, given the above and to make our methods more general, we treat image sequence frames *separately* at all stages of image processing. As such, it is possible to implement parallelization on the frame level with simultaneous multi-threading enabled to speed up the processing.

To achieve the above goals, it was decided to separate the bubble segmentation into two stages: *global* and *local* filtering/segmentation. That is, one starts by obtaining first estimates for bubble segments within images (the entire cropped FOV), and then uses a separate routine for local filtering/segmentation about the preliminary segments to improve shape detection precision at local scales (i.e., lower-wavelength corrections to first shape estimates) and to resolve false positives.

The global filtering/segmentation routine, which is essentially a completely overhauled version of our previous image processing approach [8,43], was designed and adjusted to maximize bubble detection rates at the cost of reduced shape detection precision and higher false positive rates. Further, to improve edge detection stability, we have opted for *implicit* edge detection. Global noise filtering is performed in multiple stages, each targeting a certain noise type and/or wavelength range.

As for the local filtering/segmentation routine, a recursive multi-scale analysis algorithm was implemented that is shown to perform well even for images with especially low SNR and recovers bubble shapes in cases where the first segment estimation outright fails to capture initial bubbles shapes within an acceptable margin of error. The overall structure of the proposed image processing solution is outlined in Algorithm 1.

Algorithm 1: Overall structure of the new image processing pipeline

---

   Input: Raw image sequence
1. Pre-process images (Algorithm 2)
2. Perform *global* filtering for all images (Algorithm 3)
3. Process images with the *multi-scale recursive interrogation filter* (MRIF, Algorithm 4) equipped with a *local* filter (Algorithm 5)
4. Apply the luminance map-based false positive filter to the MRIF output (Algorithm 6)
5. Apply logical filters

   Output: Centroids and shape parameters for bubbles detected in each image

---

3.2. Pre-Processing

Image pre-processing is performed in *ImageJ* as shown in Algorithm 2.

Images are cropped as indicated in Figure 1a,b. Dark current compensation is performed by subtracting the mean projection of 5 K–10 K recorded dark current noise images from all images in the raw image sequence (typically 3 K–9 K images). Then the dark current-compensated flat field is computed from the mean projection of 5 K–10 K neutron beam flux distribution images. Afterwards the dark current-compensated FOV images are normalized with respect to the flat field compensated for the dark current. These corrections for the cropped raw FOV images can be expressed as

$$I = \frac{I_0 - \langle I_{dark} \rangle}{\langle I_{beam} \rangle - \langle I_{dark} \rangle} \tag{1}$$

where I are the luminance maps of the corrected images, I_0 are the cropped raw images, and I_{dark} and I_{beam} are the dark current and neutron beam flux images. This correction results in the spatial dependence of the SNR as a consequence of less neutrons detected behind the sample compared to the open beam.

Algorithm 2: Pre-processing for raw images

　Input: Raw image sequence: 16-bit 1-channel TIFFs, 1024 × 1024 pixels
1　Crop the FOV to the liquid metal domain (Figure 1)
2　Compensate the images for camera dark current (mean)
3　Compensate the flat field images for dark current (mean)
4　Perform flat field correction (32-bit precision)
5　Convert to 16-bit
6　Remove outlying bright luminance values (median thresholding)
7　Normalize the resulting images (pixel luminance re-scaled to [0; 1])
　Output: Dark current and flat-field corrected normalized images

---

Afterwards, bright outliers are removed using median thresholding with a 1- to 2-pixel radius. The threshold is chosen such that outlier removal modifies only the pixels where luminance by far exceeds the local median (within the designated radius). Pixels with luminance above the threshold are then assigned the local median values. Finally, the images are normalized and saved, then passed to *Wolfram Mathematica* for further processing.

3.3. First Segment Estimates

First segment estimates are obtained using an algorithm referred to in this article as the *global filter*, which is outlined in Algorithm 3. Segment estimation is performed in three stages—noise filtering and background removal, implicit edge detection and segment filling and cleanup.

Noise filtering is done as follows. First, Perona-Malik (PM) filtering is performed on an input image whereby the following non-linear diffusion partial differential Equation (PDE) is solved over the image luminance map I over virtual time t with Neumann boundary conditions (BCs) [60]:

$$\frac{\partial I}{\partial t} = \nabla \cdot (f(I, \alpha) \cdot \nabla I); \ \ f(I, \alpha) = \exp\left(-|\nabla_\sigma I|^2 / \alpha^2\right) \tag{2}$$

where $I = I(\vec{r}, t)$, $f(I, \alpha)$ is the diffusion coefficient and α is its control parameter, and $I_0(\vec{r}) = I(\vec{r}, 0)$ is the input image. $f(I, \alpha)$ prescribes anisotropic diffusion by restricting luminance diffusion across sharp edges. The number of PM iterations, α and the gradient Gaussian regularization width σ that controls the sensitivity of $f(I, \alpha)$ to noise are chosen such that the PM process only acts on very small-scale noise structures—Equation (2) is edge-preserving—and it is mainly aimed at removing the remainder of the salt and pepper noise due to bright and dark outliers that survived pre-processing (Algorithm 2). Note that

$$\nabla_\sigma I = \nabla(K_\sigma \circledast I) \tag{3}$$

where K_σ is a Gaussian kernel with its standard deviation σ. In the case of our images, we found that it is productive to set α to the 0.5 quantile of $|\nabla I|$ in $I_0(\vec{r})$, and to set $\sigma = 1$ and the number of PM iterations to 5 [61,62].

Next, the total variation (TV) filter is applied assuming Poisson noise (typical for low-SNR underexposed images as in this case) by solving the following PDE with Neumann BCs [63]:

$$\frac{\partial I}{\partial t} = \underbrace{\nabla\left(\frac{\nabla I}{|\nabla I|}\right)}_{\text{Noise filtering}} + \underbrace{\frac{1}{\beta I}(I_0 - I)}_{\text{Input/output similarity}} \tag{4}$$

where the regularization parameter β determines the balance between noise filtering (pixel value variation minimization) and input/output similarity preservation. Solving (4) over virtual time t asymptotically transforms the input image into a stationary (with respect to t) filtered image. The TV filter is set up such that it eliminates noise structures up to a fraction of the length scale of bubbles while avoiding overly distorting the luminance map. In our case, we find that setting $\beta \in [0.8, 1]$ and limiting the number of TV iterations to 100 [64] yields better results.

Algorithm 3: Global first segment estimator for pre-processed FOV images

---

Input: Normalized pre-processed images (Algorithm 2)
Noise filtering and background removal

1　　Perona-Malik (PM) filter (2)
2　　Total variation (TV) filtering, Poisson model (4)
3　　Self-snakes curvature flow (SSCF) filter (5); *memoize output*
4　　Soft color tone map masking (SCTMM) (7); *memoize output*
Implicit edge detection
5　　Compute the luminance using the gradient filter (Gaussian regularization + Bessel derivative kernel)
6　　2-threshold hysteresis binarization
7　　(Optional) Morphological erosion
8　　Thinning transform
Segment filling and cleanup
9　　Filling transform
10　　Mean filtering (small radius)
11　　Otsu binarization
12　　Remove border components
Output:
- Image mask with first segment estimates
- Memoized SSCF-filtered image for later use in Algorithm 4
- Memoized SCTMM-filtered image for later use in Algorithm 6

The third stage is the application of the self-snakes curvature flow (SSCF) filter. Here the following PDE is solved with Neumann BCs [65]:

$$\frac{\partial I}{\partial t} = |\nabla I| \cdot \nabla\left(g(I, \gamma) \cdot \frac{\nabla I}{|\nabla I|}\right); \ \ g(I, \gamma) = \exp\left(-|\nabla I|^2/\gamma^2\right) \tag{5}$$

where $g(I, \gamma)$ is the curvature diffusion coefficient and γ is its control parameter. Unlike the PM filter (2), SSCF diffuses local luminance curvature, not luminance itself. It is also edge-preserving. The number of SSCF iterations and γ are set such that the SSCF filter eliminates any remaining sharply localized luminance maxima about the bubbles since there the luminance curvature is the greatest; at the same time, filtering must preserve bubble region contrast with respect to background. For our images we find that in some cases one may set $\gamma \to \infty$, which simplifies (5) to

$$\frac{\partial I}{\partial t} = |\nabla I| \cdot \nabla\left(\frac{\nabla I}{|\nabla I|}\right) \tag{6}$$

which is the mean curvature flow PDE. However, other cases require anistropic curvature diffusion and then we set γ to the 0.5 quantile of $|\nabla I|$ in $I_0(\vec{r})$ as with the PM filtering. The number of performed SSCF iterations is set to 7 [66].

The next stage is the soft color tone map (CTM) masking (SCTMM) which is a non-linear filter designed to clean up the image background by removing large-scale artefacts left over after denoising and to further separate background from bubbles while avoiding excessive erosion of the bubble regions. The large-scale structures in the background were actually one of the sources of the edge detection instability in the previous approach, especially for low-CNR images (CNR $= |S_b - S_0|/\sigma_n$; S_0—background signal intensity) with higher bubble number density where bubble detection was often outright impossible due formed edge artefacts that could not be reliably removed.

Given a normalized original image x, the SCTMM background correction generates a new image y:

$$y = x * \left(\underbrace{x - \underbrace{(1 - \text{CTM}(x,c))}_{\text{Soft thresholding}}}_{\text{Soft background mask}} \right) \qquad (7)$$

where $\text{CTM}(x,c)$ is the CTM operation and c is the luminance compression factor. The CTM operation maps the colors (in this case the gray-scale values) of the image using gamma compression with a global compression factor c [67]. The idea behind SCTMM (7) is as follows. A pure $x * x$ product would have the effect of non-uniformly increasing the distances between the nearest luminance values (input x is normalized) and luminance maxima and minima values would be much more distant from the mid-range luminance values, which are affected the most. If one masks or lowers the values of certain pixels within one of the images x', then $x * x'$ would act as a "*soft*", i.e., weighed, mask (as opposed to a "hard" binary mask) that would shift the pixels with reduced values in x' further towards the lower end of the luminance range, ideally making them background. Soft masking is preferred here because masking using binarization and then replacing the removed background using luminance interpolation or other methods will generally produce artificial and potentially very pronounced edges and/or reduce the contrast of actual edges.

Here it is required that x' is such that background and post-filtering artefacts are removed, and bubble contrast is enhanced while bubble features are eroded as little as possible. It was decided to opt for additive masking of the form $x'(x) = x - mask$. An inverted $\text{CTM}(x,c)$ was chosen as *mask* because, if the right c value is set, $1 - \text{CTM}(x,c)$ will have high luminance for background and denoising artefacts since $\text{CTM}(x,c)$ reduces the difference in their luminance values. This way $x' = x - (1 - \text{CTM}(x,c))$ has, conversely, greatly reduced luminance for artefacts and background. The resulting product (7) then has the desired properties and emphasizes the bubbles while reducing the impact of image artefacts. For the images considered here $c = 0.5$ generated better results.

Another major change is the substitution of explicit segmentation and edge detection with an implicit procedure. The SCTMM output is normalized and passed to the gradient filter (luminance gradient magnitude map of a Gaussian kernel convolution (3) for an image; the Bessel derivative kernel is used) which estimates bubble edge regions (halos) in the image. The halos are segmented using double-threshold hysteresis binarization (pixel corner connections enabled) [68] and then the edge estimates are obtained using the thinning transform [68] (exploiting the edge gradient symmetry). Optionally, morphological erosion [69] can be applied to the halo segments before thinning, which is suggested if thinning outputs jagged edges. Finally, bubble shape masks are generated by applying the filling transform [68] followed by a small-radius mean filtering. The advantage of this procedure is that it is more stable and does not require edge/area repairs for bubble edges/masks.

Afterwards, small-radius (2–3 pixels) mean filtering step followed by Otsu binarization [70] ensures that the edge dendrites that are occasionally left over at the filled bubble shape boundary are pruned—this is much computationally cheaper than morphological pruning, which would also generally require multiple iterations to converge to edges that

are Jordan curves immune to the pruning transform. The gradient filter regularization kernel scale is set to a fraction of the bubble length scale (Section 2.3). The primary threshold for the hysteresis binarization of edge halos is by default 0.35, though it had to be reduced to 0.25 in some cases where the image quality was especially problematic. The secondary threshold is computed using the Otsu method. In our case we perform morphological erosion using disk structural elements with a 5-pixel radius. Finally, we remove segments that are in contact with image boundaries—this is because only bubbles that are fully within the FOV are of interest.

3.4. Multi-Scale Recursive Interrogation Filter (MRIF)

3.4.1. MRIF Core

The filtering methods utilized in Algorithm 3 are rather aggressive. Testing revealed that, while bubble detection rates are indeed significantly higher than before and bubbles are detected everywhere within the FOV, it comes at the cost of decreased shape resolution precision and a higher false positive rate. The former is very important for a more in-depth analysis of the effects of varying flow rate and MF on the behavior of bubble chains. It was decided to fine-tune the global filter such that the bubble detection rates are maximized and good first estimates of bubble shapes/sizes are obtained, and complement it with a routine that would use the first estimates to generate more precise bubble shapes and efficiently filter out false positives. To this end, we have developed an algorithm for iterative segmentation refinement—the *multi-scale recursive interrogation filter* (MRIF) outlined in Algorithm 4 and schematically illustrated in Figure 4.

Algorithm 4: Multi-scale recursive interrogation filter (MRIF)

Input:
- An image mask with first segment estimates (Algorithm 3, Step 12)
- The global SSCF filter output (Algorithm 3, Step 3)

1 Define square (side length L') interrogation windows (IWs) for segments based on their areas and centroids (8) and (9)
2 Define the FOV image as an IW with scale L based on FOV dimensions
3 *For every initial segment and updated segments:*

    while $L'/L < \varepsilon,\ \ \varepsilon > 1$ *(user-defined)* **do**
        Map the global SSCF filter output onto the segment IW
        Perform local filtering (Algorithm 5)
        if *updated segments were found* **then**
            Define new IWs (side length L') for the updated segments based on their areas and centroids (8) and (9)
            Redefine preceding IW scales as $L' \rightarrow L$
        else
            Break
        end
    end

4 *Memoize* converged IWs for all segments
5 Map the centroid coordinates of the resulting segments from converged IWs onto the original image
6 Map the updated segment masks onto the original image and build an updated global mask

Output:
- Updated bubble shape masks for the FOV
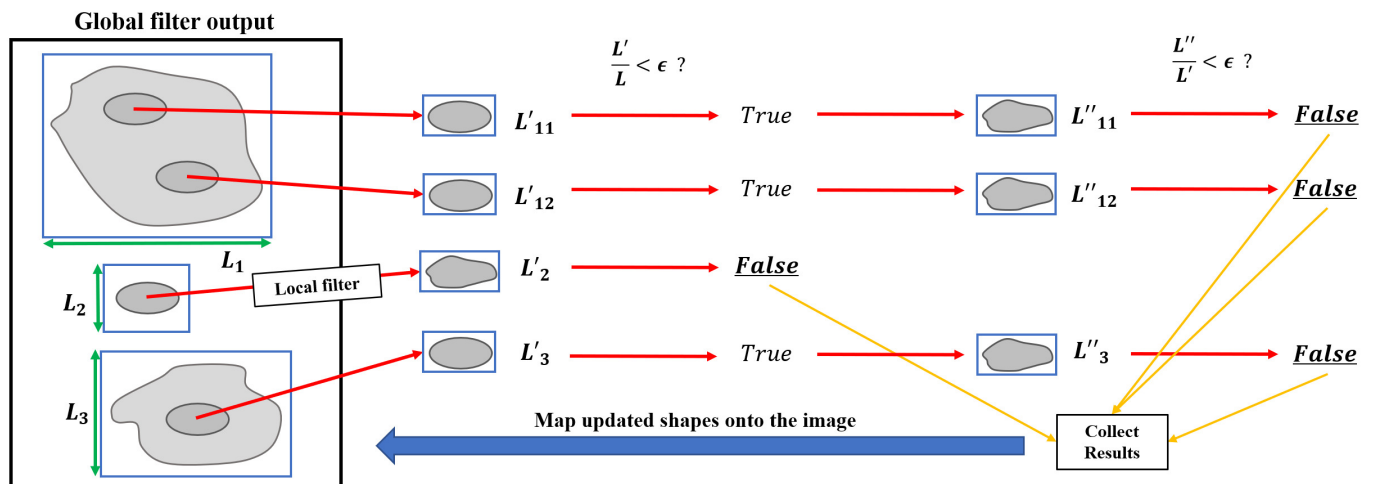- Converged IWs for later use in Algorithm 6

Figure 4. A schematic illustration for the MRIF algorithm. The gray domains represent detected segments.

The key idea is to define *interrogation windows* (IWs) about the initially detected bubbles to exclude irrelevant parts of the image and its intensity histogram from the analysis. This also helps to reduce the influence of any remaining artefacts left over from the global filtering. Especially for images with lower SNR and with closely packed bubbles, it may be the case that the initial segmentation is poor, i.e., two or more bubbles have been segmented as one due to the surrounding artefacts, or a bubble was connected to large-scale artifact structures, forming a large segment that obscures the true object. This means that an appropriate local filtering algorithm must be devised for IWs. However, a single local filtering pass may not be enough for various reasons, e.g., what might appear visually as a poorly segmented bubble at the scale of the current IW might actually turn out to be, after local filtering, multiple bubbles—these would then each require another pass at a finer scale. This means that in general a series of consecutive interrogation passes could take place. Thus, MRIF performs object filtering at different scales, effectively checking that the segments have been properly resolved by the global filter and/or in the preceding local filtering iterations. A stopping criterion based on the IW size similarity between iterations makes sure that MRIF recognizes that it only makes sense to re-filter an image patch if the object is significantly smaller than the previous IW, since in this case the finer shape features may have been under-resolved.

MRIF consists of several components: an IW generator that centers the IW at the segment location and adjusts its size according to the segment size; a local filter that is responsible for filtering within IWs; a recursive routine that performs a "scale descent" and converges to the "true" segment scale starting from the first estimates; a procedure that collects the final, updates segments and maps them onto the original bubble segment mask, substituting the first estimates. The ε criterion is the stopping factor that controls recursion depth, i.e., the lower IW scale threshold. In our case $\varepsilon = 2$ yielded good results. The MRIF core components are described in detail in Sections 3.4.2–3.4.4.

3.4.2. Interrogation Windows (IWs)

IWs are square windows with side length L which is determined for segments as follows:

$$L = \lfloor s \cdot \sqrt{S/\pi} \rceil \tag{8}$$

where S is the segment area in pixels and s is a user-defined scale factor, i.e., L scales with the equivalent radii of segments. Given L and the segment centroid $\vec{r} = (x, y)$, an IW is defined by pixel coordinate intervals:

$$\text{IW} : \left\{ \left[\lfloor x \rceil - \frac{L}{2}; \lfloor x \rceil + \frac{L}{2} \right], \left[\lfloor y \rceil - \frac{L}{2}; \lfloor y \rceil + \frac{L}{2} \right] \right\} \tag{9}$$

Since L for the entire image is required for the first iteration of MRIF and one should always perform local filtering at least once for every initial segment, here one can set $L = s_0 \cdot \max \dim(z)$ where z is the SCCM-filtered FOV image (Algrithm 3, Step 3) and $s_0 \geq 1$ is an arbitrary scaling factor such that $L'/L > \varepsilon$ is always true for the first MRIF iteration.

Note that in general, an IW may be out of the image bounds—for this reason, two IWs are generated for a segment at every MRIF iteration: a *virtual IW* defined by (9), and a *real IW* given by

$$IW' = z \cap IW \tag{10}$$

so IW' is not necessarily square. To reiterate: the current filtering scale is defined by the IW scale L and the SSCF filter output is mapped onto IW'. For simplicity, IW' will be referred to as IW from this point, as in Algorithm 4, unless stated otherwise later. We observed that for our images $s \in [2; 2.5]$ yielded better results.

3.4.3. Local Segment Updates

The local filter used for recursive filtering in MRIF (Algorithm 4, Step 3) has elements similar to the global filter, but with important distinctions. It was designed to be less aggressive because MRIF ensures that only the crucial background context from the image is retained within an IW, e.g., the somewhat destructive SCTMM is not required. The local filter operates on an IW as described in Algorithm 5.

Here the mean filtering radius is a fraction of the expected bubble scale—this is to avoid averaging out the finer shape features. The filtering radius is set based on the expected lower threshold for bubble surface perturbation wavelength. The gradient filter regularization kernel (3) scale was set equal to the mean filtering radius.

Algorithm 5: Local segment refinement for IWs

Input: Global SSCF filter output (Algorithm 3) mapped onto an IW
1　Mean filtering
　　Implicit edge detection
2　Compute edge halos using the gradient filter (Gaussian regularization + Bessel derivative kernel)
3　Chan–Vese binarization (11)
4　(Optional) Morphological erosion
5　Thinning transform
　　Segment filling and cleanup
6　Filling transform
7　Mean filtering (small radius)
8　Otsu binarization
9　Remove border components
Output: Updated local shape estimate

Chan–Vese binarization is a variational method for object segmentation in images that does not explicitly utilize edges. It is generally more robust than edge- and histogram-based methods (e.g., Otsu, Canny [70,71]) but is more computationally expensive [72]. However, this can be afforded relatively easily in the case of IWs that are only a fraction of the original images. Chan–Vese two-level segmentation works by assigning the following functional to an image (in this case single-channel) and minimizing it iteratively [72]:

$$\min \underbrace{\mu \cdot L(C)}_{1} + \underbrace{\nu \cdot S(C)}_{2} + \underbrace{\lambda_1 \int_{D_1} \left| I(\vec{r}) - \langle I \rangle_{D_1} \right|^2 dS}_{3} + \underbrace{\lambda_2 \int_{D_2} \left| I(\vec{r}) - \langle I \rangle_{D_2} \right|^2 dS}_{4}; \ C = \partial D_{1,2} \tag{11}$$

with respect to C, where C is the set of segment contours, L is the length of C, S is the area enclosed by C, and μ, ν, λ_1 and λ_2 are the control parameters. The Chan–Vese process is typically initialized by defining C such that the image area is covered with a checkerboard of small circular contours of adjustable size, preferably very fine [72]. Two regions within an image are defined: areas within C initially given by circular contours (D_1) and outside C (D_2). Minimizing (11) has several effects on C: their combined length (term 1), area of D_1 (term 2), as well as the total discrepancy between the region luminance values and the region averages (terms 3 and 4) are reduced—the relative prevalence of these effects is dictated by μ, ν, λ_1 and λ_2. C is defined as the zero crossing of a special level set function [72]. Optimization is performed for a certain number of iterations, generally resulting in the unification/dissolution of the initial disjoint C components until, ideally, C encapsulates the desired segments in the image. The Chan–Vese process also guarantees that C is a set of Jordan curves.

Minimization of (11) can be performed by solving the following PDE with respect to the level set function φ for C [72]:

$$\frac{\partial \varphi}{\partial t} = \delta(\epsilon, \varphi) \left(\mu \cdot \nabla \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) - \nu - \lambda_1 \left(I - \langle I \rangle_{D_1} \right)^2 + \lambda_2 \left(I - \langle I \rangle_{D_2} \right)^2 \right) \qquad (12)$$

with the BC

$$\frac{\delta(\epsilon, \varphi)}{|\nabla \varphi|} \cdot \frac{\partial \varphi}{\partial \vec{n}} = 0 \qquad (13)$$

defined for $\partial D : D = D_1 \cup D_2$. Here $\varphi = \varphi(\vec{r}, t)$ with $|\varphi| = 1$, $\vec{n}$ is the outward boundary normal, and $\delta(\epsilon, \varphi)$ is the level set regularization function

$$\delta(\epsilon, \varphi) = \frac{\epsilon}{\pi(\epsilon^2 + \varphi^2)} \qquad (14)$$

$\varphi(\vec{r}, t)$ is initialized as

$$\varphi(\vec{r}, 0) = \sin \left(\frac{\pi x}{\lambda_\varphi} \right) \cdot \sin \left(\frac{\pi y}{\lambda_\varphi} \right) \qquad (15)$$

where λ_φ determines the wavelength of the initialized checkerboard pattern. Equation (12) with (13) and (15) is then solved iteratively by alternating between updates for $\langle I \rangle_{D_1}$ and $\langle I \rangle_{D_2}$, and φ.

In our case $\mu = 0.03$, $\nu = 0$, $\lambda_1 = \lambda_2 = 1$ and $\epsilon = 1$. As indicated in [72], (15) with $\lambda_\varphi = 5$ is a good initialization strategy leading to fast convergence, but we have observed that $\lambda_\varphi \in [2; 5)$ yields faster convergence without perceivable quality degradation in our cases. The reason Chan–Vese instead of Otsu or hysteresis binarization is used is that the former exhibits much stabler edge halo detection in IWs and is not explicitly tied to image histograms (i.e., does not *only* minimize inter-class and maximize intra-class variance for the level sets like the Otsu method) which can be very different across IWs. Once binarization is complete, one performs the same sequence of operations as with the global filter (Algorithm 3, Steps 7–12). For local filtering, erosion for segments is performed with 1- to 5-pixel radius disk elements and the mean filtering (cleanup) radius is 1–3 pixels.

3.4.4. Mapping Updated Segments to Original Images

Centroids are mapped from converged IWs onto original images (full FOV) using the following transformation:

$$\vec{r}\,' = \vec{r}_{\text{iw}} + (\vec{r} - \vec{r}_0(L)) - \vec{\Lambda}(L, \vec{r}_{\text{iw}}) \qquad (16)$$

where $\vec{r}\,'$ is the updated centroid location in the FOV, $\vec{r}$ are the updated centroid coordinates from MRIF in the IW coordinate system, $\vec{r}_{\text{IW}}$ is the location of the virtual IW center in the FOV, $\vec{r}_0(L)$ is the center of the virtual IW in its coordinate system and $\vec{\Lambda}(L, \vec{r}_{\text{iw}})$ is the

IW crop correction for $\vec{r}_0(L)$. The mapping is illustrated in Figure 5. Since IW and IW′ both always contain the segment (if any detected) and their centers are always within the FOV, for an IW fully within the bounds of the FOV one simply extends a radius vector $\vec{r}_{iw}$ (stored by MRIF, Algorithm 4) to the IW center and then from that to the bubble centroid via $\vec{r} - \vec{r}_0(L)$ (Figure 5a, the red vector). If the virtual IW is partially out of bounds, then the actual IW′ (Figure 5b) has a different center coordinate $\vec{r}_0{}' = \vec{r}_0 + \vec{\Lambda}(L, \vec{r}_{iw})$ since it is displaced when the IW is cropped (10). The crop correction given an image z is as follows:

$$\vec{\Lambda}(L, \vec{r}_{iw}) = \vec{e}_x \cdot \begin{cases} x_{\min}/2, & x_{\min} < 0 \\ 0, & x_{\min} \geq 0 \end{cases} + \vec{e}_x \cdot \begin{cases} 0, & x_{\max} \leq w \\ -(x_{\max} - w)/2, & x_{\max} > w \end{cases}$$
$$+ \vec{e}_y \cdot \begin{cases} y_{\min}/2, & y_{\min} < 0 \\ 0, & y_{\min} \geq 0 \end{cases} + \vec{e}_y \cdot \begin{cases} 0, & y_{\max} \leq h \\ -(y_{\max} - h)/2, & y_{\max} > h \end{cases} \quad (17)$$

where $x_{\min}$, $x_{\max}$, $y_{\min}$ and $y_{\min}$ are given by L and $\vec{r}_{iw}$ via (9) and $\dim(z) = (w, h)$. The IW crop correction is required as a separate step because the IW coordinate system origin is not necessarily within the FOV.
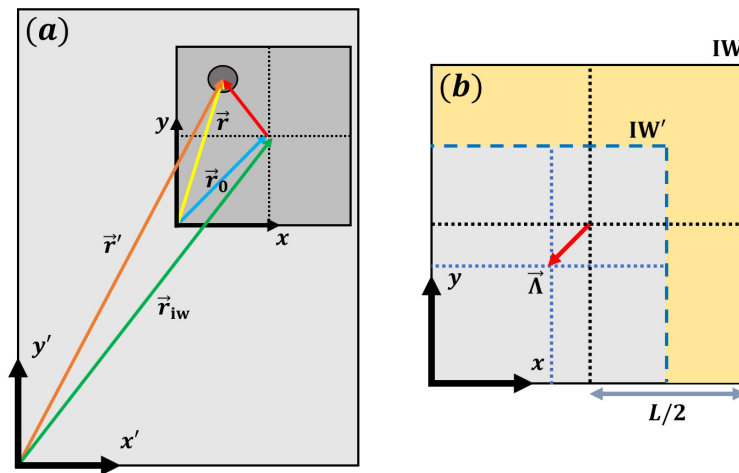

Figure 5. Coordinate mapping from IWs onto the full FOV for objects detected in IWs: (**a**) transformation from the IW coordinate system to the image coordinate system and (**b**) crop correction for the IW center coordinates within the FOV. In (**a**) the circle represents the object detected in the IW (dark gray square). In (**b**) the yellow area indicates the out-of-bounds part of the virtual IW.

3.5. Luminance Map-Based False Positive Filtering

The MRIF procedure, in addition to its intrinsic false positive filtering capacity stemming from the recursive multi-scale analysis, is supplemented by a dedicated false positive identification algorithm that processes the segments output by MRIF. The procedure is outlined in Algorithm 6.

Algorithm 6 exploits the observation that SCTMM applied locally to the global SSCF filter output mapped onto converged IWs should produce very strong intensity maxima and an overall higher intensity in the segment region in the case of a true bubble while the opposite should hold for false positives. The product of mean and maximum intensity is used so that neither of the two criteria alone are enough to pass the filter, since it might be the case that a region with an otherwise background level intensity might exhibit a tightly localized intensity maximum; similarly, mean intensity filtering alone is not enough since a bubble should have a strong maximum of transmission intensity about its centroid, which in itself is not as strongly correlated with the mean intensity. Another way to interpret this is that, if the maximum and mean thresholding have certain probabilities of accepting a false positive, then the max/mean product thresholding has a false positive acceptance probability at least lower than the greater of the two components. Note that Algorithm 6

uses a luminance compression factor $c = 0.5$ for SCTMM. We found that false positives are efficiently filtered (i.e., also minimizing true positive elimination) when $\eta \in [0.1; 0.125]$.

Algorithm 6: Luminance map-based false positive filtering for MRIF output

---

    Input:
-    Updated bubble shape masks for the FOV (Algorithm 4)
-    Global SCTMM filter output (Algorithm 3, Step 4)
-    Converged IWs (Algorithm 4)

1 Map the global SCTMM filter output onto converged IWs
2 Apply the SCTMM correction (7)
3 Multiply by the segment binary mask
4 Normalize the image
5 Compute $\langle I \rangle \cdot \max(I)$ for all segment regions
6 *Thresholding for all segments:*

   if $\langle I \rangle \cdot \max(I) < \eta; \ \ \eta \in [0; 1]$ *(user-defined)* **then**
   | Flag the segment as a false positive
   else
   | Nothing
   end

7 Remove the identified false positives from the bubble shape mask
   Output: FOV bubble shape masks without detected false positives

---

After filtering the MRIF output using Algorithm 6, all properties of interest are measured for all remaining bubble segments and logical filters can be applied for further false positive elimination. In our case, logical filters check for implausible bubble coordinates, sizes and aspect ratios and remove the outliers from the dataset of measured bubble shapes. Finally, the resulting data can be post-processed and interpreted.

4. Performance Analysis

4.1. First Segment Estimates

Figure 6 shows the effect of subsequent operations that the global filtering routine (Algorithm 3) performs for a pre-processed image.

The difference between Figure 6a,b is that the noise with low wavelengths (sharply localized luminance maxima and minima left over from pre-processing) have been eliminated. Next, the TV filter (Figure 6c) consolidates the high luminance values within the bubble regions (white dashed circles), increasing the SNR for the bubbles. Furthermore, noise is significantly damped and the wavelength of its features is increased even more. However, the TV filter does not remove the sparse larger-scale luminance maxima still seen in Figure 6c in the background between the bubbles as efficiently as it is desired without degrading the CNR for the bubbles. This function is performed by SSCF (Figure 6c) which specifically diffuses the leftover intensity maxima, increasing bubble SNR (and CNR due to reduced noise in the background) further. In the case of Figure 6 one has $\gamma \to \infty$ for (5). A close-up view of one of the bubble neighborhood's transformations is shown in Figure 7 where one can visually notice that bubble CNR is increased by SSCF (d) and that intensity maxima are indeed eliminated from the background. A more detailed analysis of how how noise filtering stages affect the image and the bubbles can be seen in Figure 8 where a vertical strip containing both bubbles is taken from the images in Figure 6 and their relief plots (a) and mean luminance profiles (b and c) are shown.
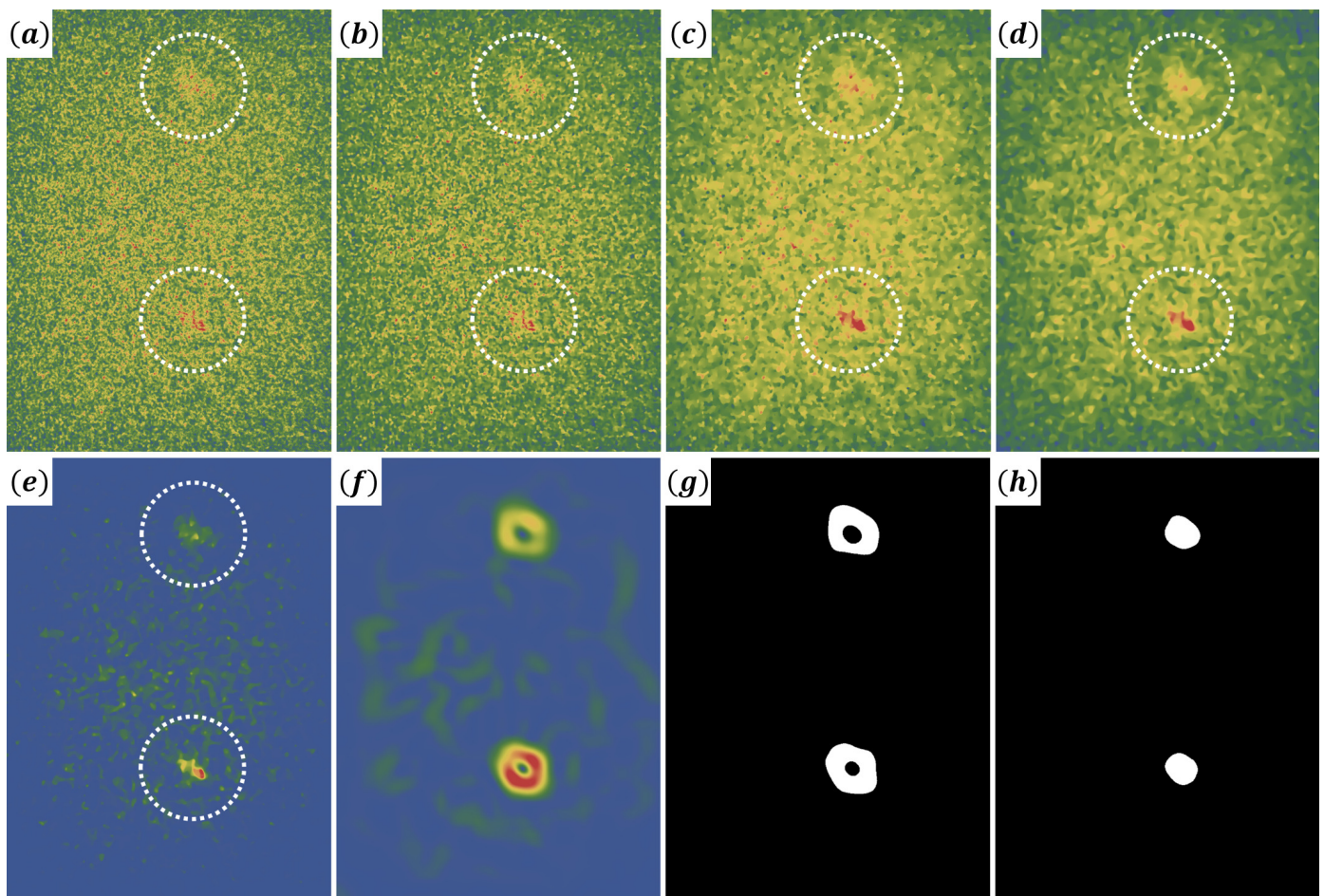
Figure 6. First segment estimation stages (Algorithm 3): (**a**) original pre-processed (Algorithm 2) image; (**b**) PM-filtered image; (**c**) Poisson TV-filtered image; (**d**) SSCF-filtered image; (**e**) SCTMM output; (**f**) gradient filter output (edge halos); (**g**) hysteresis-segmented edge halos (prior to erosion); (**h**) bubble masks obtained after erosion, thinning, filling, small-radius mean filtering, Otsu binarization and border component removal. The color scheme is as in Figure 1.
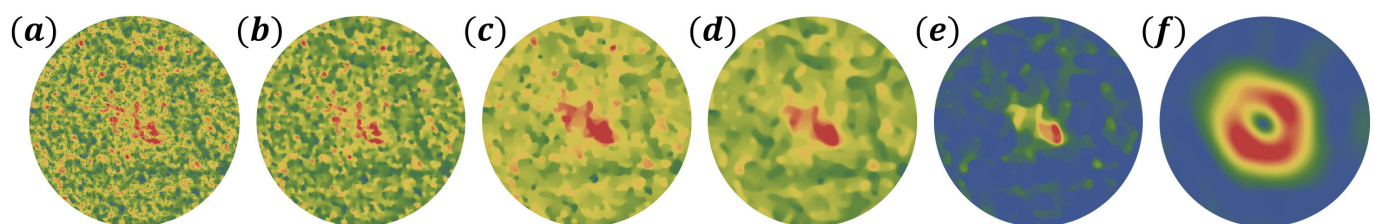

Figure 7. Effects of global filtering (Algorithm 3) on the luminance map of the neighborhood of the lowermost bubble in Figure 7 indicated by a white dashed circle: (**a**) neighborhood projection of the original pre-processed image; (**b**) PM-filtered; (**c**) Poisson TV-filtered; (**d**) SSCF-filtered; (**e**) SCTMM output; (**f**) gradient filter output (edge halos).

With SNR increased by the sequence of PM, TV and SSCF filters, SCTMM is now applied to increase the CNR—this is especially clearly seen in Figure 8c where one can see that SCTMM significantly flattens the background while preserving bubble signal intensity, as intended. This enables the gradient filter to produce bubble edge halos with an even greater CNR (note the high depth of the halo "wells" in Figure 8(a6), which extend well below the binarization threshold, down to background luminance levels), enabling clean segmentation, as seen in Figure 6g.
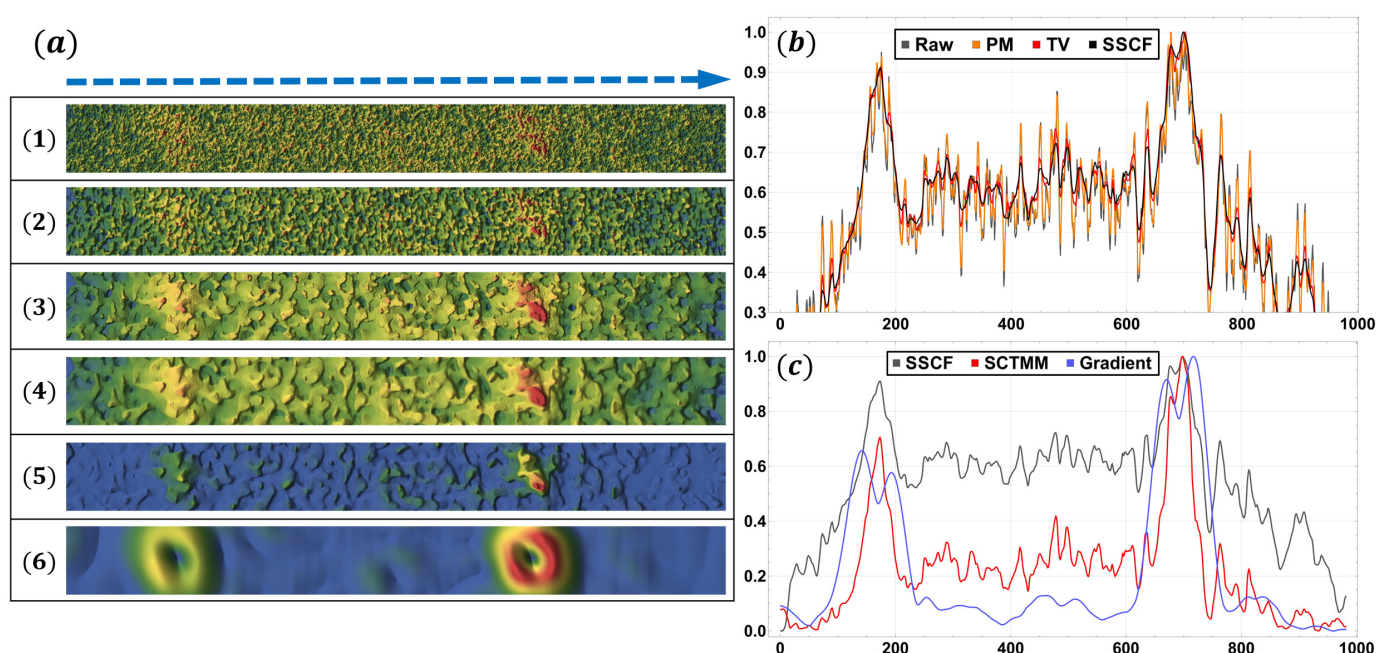
Figure 8. The effect of filtering stages on bubble signal versus noise and background: (**a**) pseudo-3D shaded relief plots of strips taken from Figure 6a–f (sub-figures 1–6, respectively) containing bubbles and background in between; (**b**) mean luminance profiles over image strips (1–4) in the direction indicated by a blue dashed arrow in (**a**) for pre-processed, PM-, TV- and SSCF-filtered images; (**c**) mean luminance profiles for SSCF-, SCTMM- and gradient-filtered strips (4–6). Note that all luminance profiles in (**b**,**c**) have been normalized for direct comparison.

4.2. MRIF

The first segment estimator could have been enough for bubble shape extraction, when tuned appropriately, if not for the fact that the image considered in Figures 6–8 is one of the better examples in terms of noise and artefacts present, i.e., a considerable portion of the images captured in our experiments are of a much poorer quality. Not only are the obtained shapes often imprecise or deformed, they can at times be decidedly non-physical—one such example is show in Figure 9, where in (a) a segment is shown that looks like two bubbles that are in the process of merging.

Such events are not expected at the flow rate for which this image was acquired, therefore Figure 9a shows an obvious artefact. The bottom-right corner of (b) contains closely packed high-luminance spots which likely have been combined and merged with the bubble region in the upper-left corner. However, once MRIF targets the segment and the local filter is applied to the SSCF output projected onto the segment IW (c) in stages (d–h), the artefact is no longer present and a single bubble is correctly resolved.

In addition to such artefacts, since the global filter was tuned to maximize the odds of detecting bubbles in the FOV, there are cases where detected segments are false positives. Two instances of such segments interrogated by MRIF are shown in Figure 10. In (a) one can see that the local filter has revealed that there is indeed no segment contained within the IW.

However, interrogating the false positives in an IW might on occasion produce segments yet again, as in (b), where the gradient filter stage (b4) generated a structure that resembles a bubble edge halo. It was then segmented and, through edge cleanup, transformed into a segment that seems eligible—but simply overlaying it over the original image projected onto the IW, one can see that this is not the case. However, MRIF effectively performs a two-factor false-positive check, and in such cases the luminance map-based filter (Algorithm 6) serves as a backup. Once the global SCTMM output is projected onto the IW and SCTMM is applied to the resulting image (b9), one can see that the segment overlay contains only background, and thus this false positive will be eliminated since it has $\langle I \rangle \cdot \max(I) < \eta$.
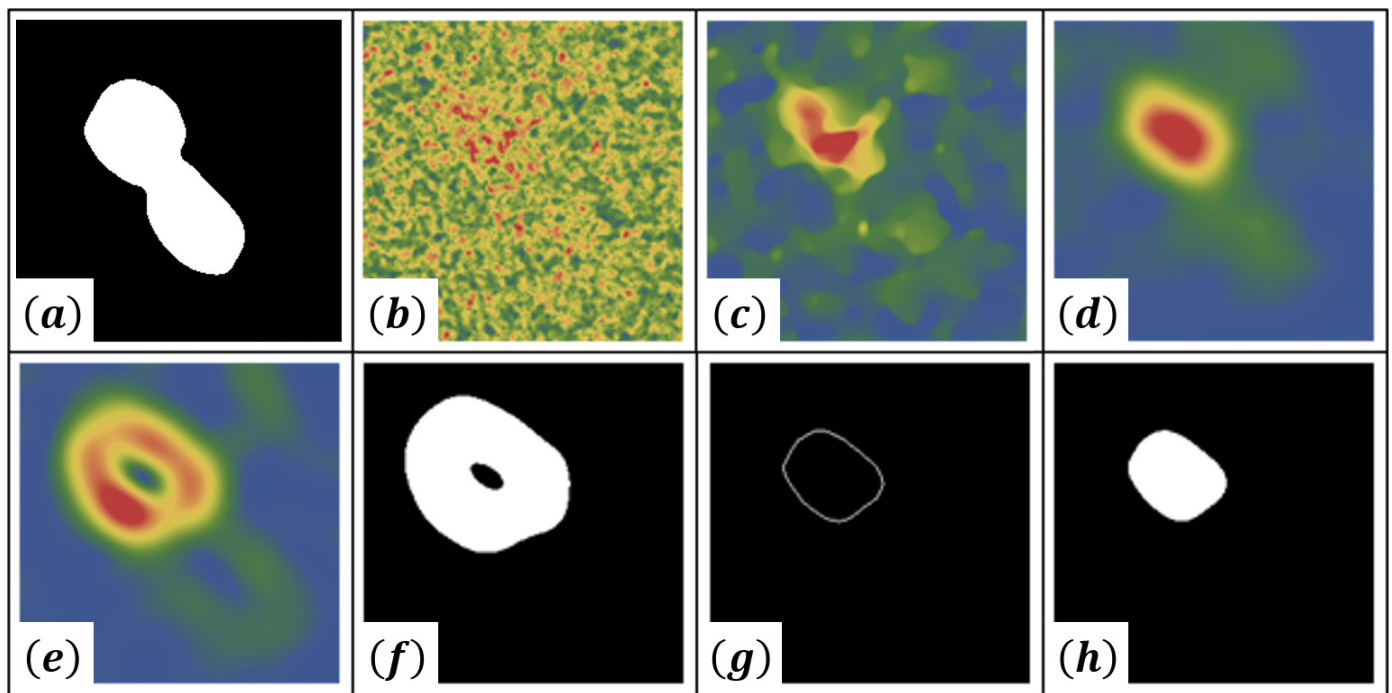
Figure 9. Updating a first segment estimate using the local filter (Algorithm 5): (**a**) first estimate in an IW; (**b**) original pre-processed image projected onto the IW; (**c**) SSCF output projection onto the IW (Algorithm 3); (**d**) mean-filtered SSCF output; (**e**) gradient filter output; (**f**) Chan–Vese segmentation output; (**g**) edges extracted via erosion and thinning; (**h**) updated local segment for the IW. Segment filling and cleanup are similar to what is done in Algorithm 3.
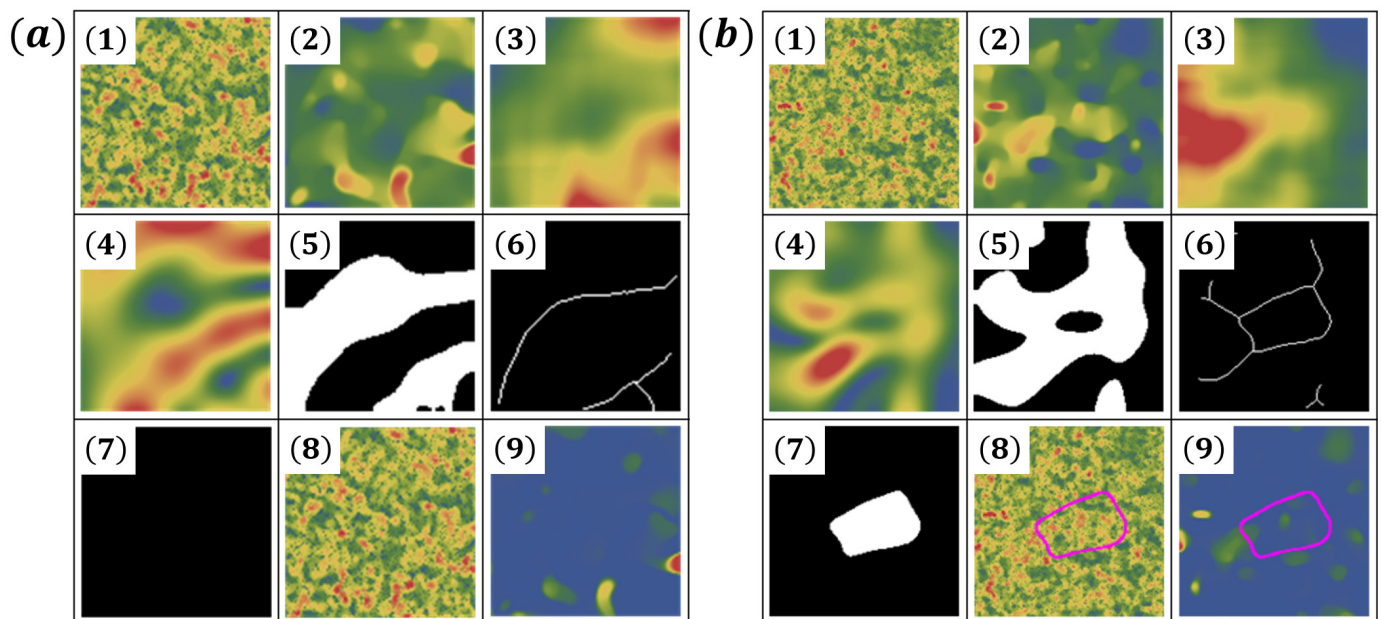

Figure 10. Instances of false positives revealed by MRIF with (**a**) no local segments found and (**b**) an artefact (purple outline) that will be eliminated later by Algorithm 6 based on the segment luminance map. Sub-figures (1–9) in (**a**,**b**) are: (1) is the original image projected onto an IW; (2–7) are the respective local filtering stages (Figure 9c–h); (8) is (1) with detected edge overlays; (9) is SCTMM applied to (2).

An example containing several instances of false positives, under-resolved shapes and bubble regions merged with noise patterns is shown in Figure 11. Notice that the image quality, even visually, is much worse than in Figures 6–8. The artefacts in the upper part of the image stem from lower CNR in (b), whereas one of the bottom artefacts comes from a

noise structure in the background that resembles a bubble. However, as seen in (d,e), MRIF successfully removes all false-positives and artefacts while improving the shape estimates.

An even more difficult case is seen in Figure 12 where very large segments appear. The largest one in the upper part of (c) has occluded two of the four bubbles visible in (a,b). This is also an instance where the crop correction (17) is significant for remapping the updated segments onto the FOV (16). MRIF successfully resolves bubbles from the first estimates, as seen in (d–f). Notice the segment in the bottom-left part of (e,f)—its luminance map contains background only, so it will be later eliminated by Algorithm 6.

To see how MRIF iteratively resolves cases like the above two examples, consider Figure 13 where the updates for the first segment estimates are shown for MRIF iterations. One can see in (e3) that the SNR and CNR are even worse than in the cases shown in Figures 11 and 12. The largest segment seen in (e1,a1) is first resolved into two bubbles (b1) and then each bubble is interrogated once more, obtaining more precise shapes. The bottom-most segment in (e1) requires the most MRIF iterations—the first two, (b3) and (c3), remove portions of the artefact that had obscured the bubble, and the last iteration (d) updates the resolved shape. The resulting bubble shapes are then mapped onto the FOV as indicated in (b–e).
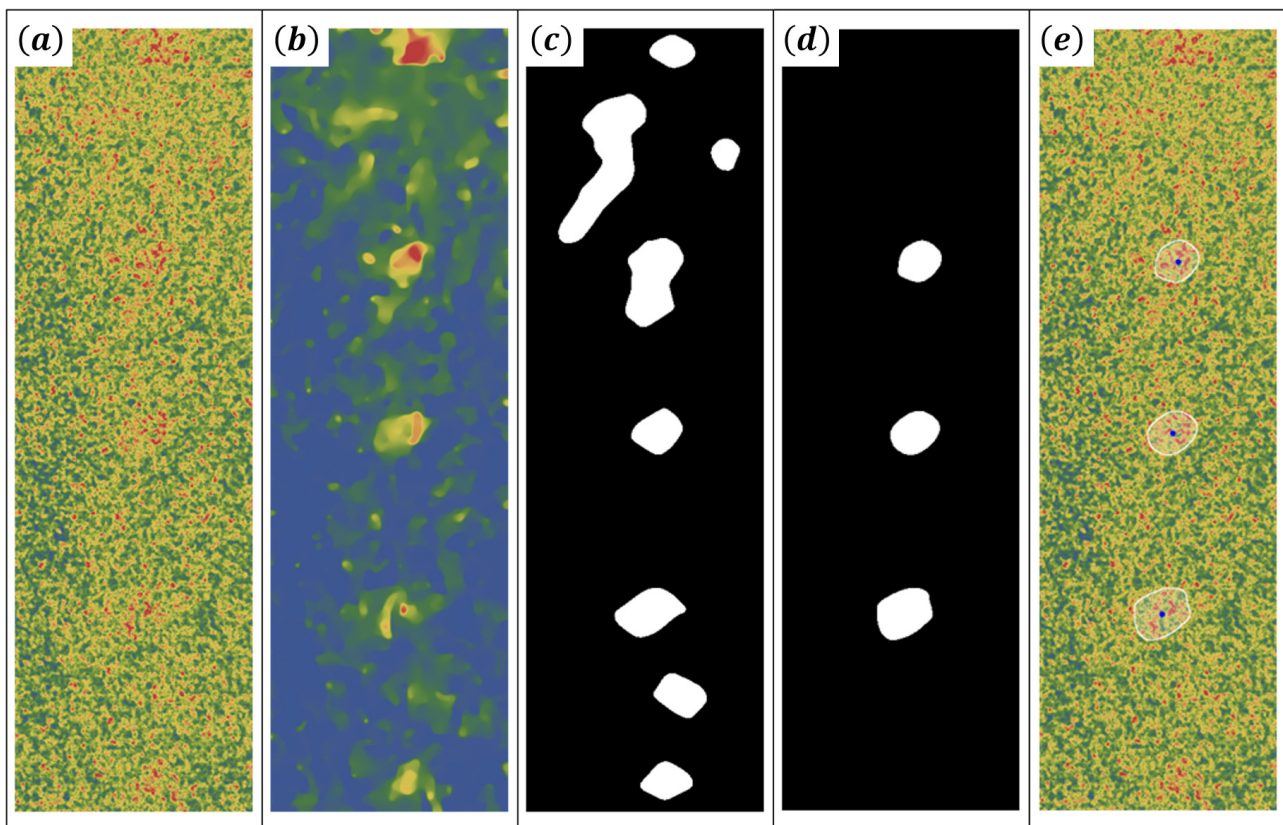

Figure 11. An example of segmentation improvement by MRIF: (**a**) horizontally cropped pre-processed image—note the very low CNR in the upper half of the image; (**b**) global SCTMM filtering result; (**c**) first segment estimates; (**d**) updated segments output by MRIF; (**e**) output segment overlays for (**a**). Note that the top- and bottom-most segments from (**c**) are not present in (**d**,**e**)—this is the correct behaviour, since one can visually see in (**a**,**b**) that the corresponding bubbles are partially outside of the FOV, and thus are not eligible for analysis.
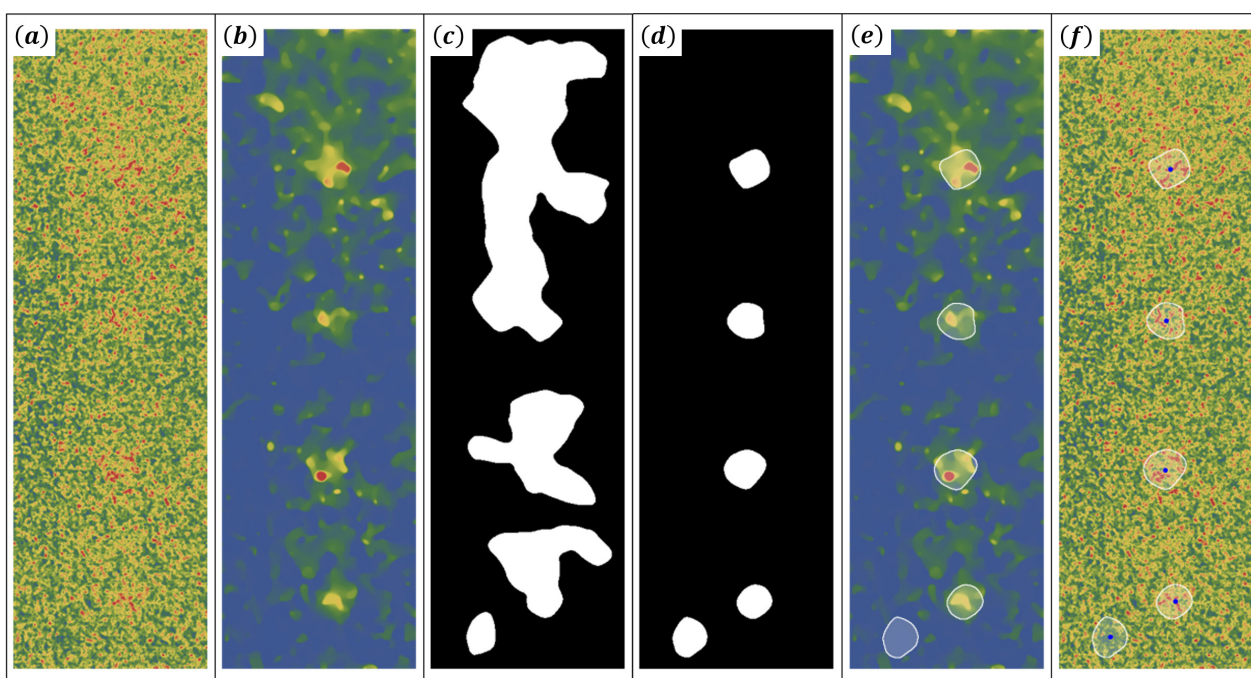
Figure 12. Another example of MRIF resolving initially occluded and incorrectly detected segments. Sub-figures (**a**–**d**,**f**) correspond to Figure 11a–e, respectively, and (**e**) shows the MRIF output overlays for the global filter output. The false positive in the bottom-left corner of (**d**–**f**) that survived MRIF interrogation will be eliminated by Algorithm 6, since it corresponds to background (**e**).
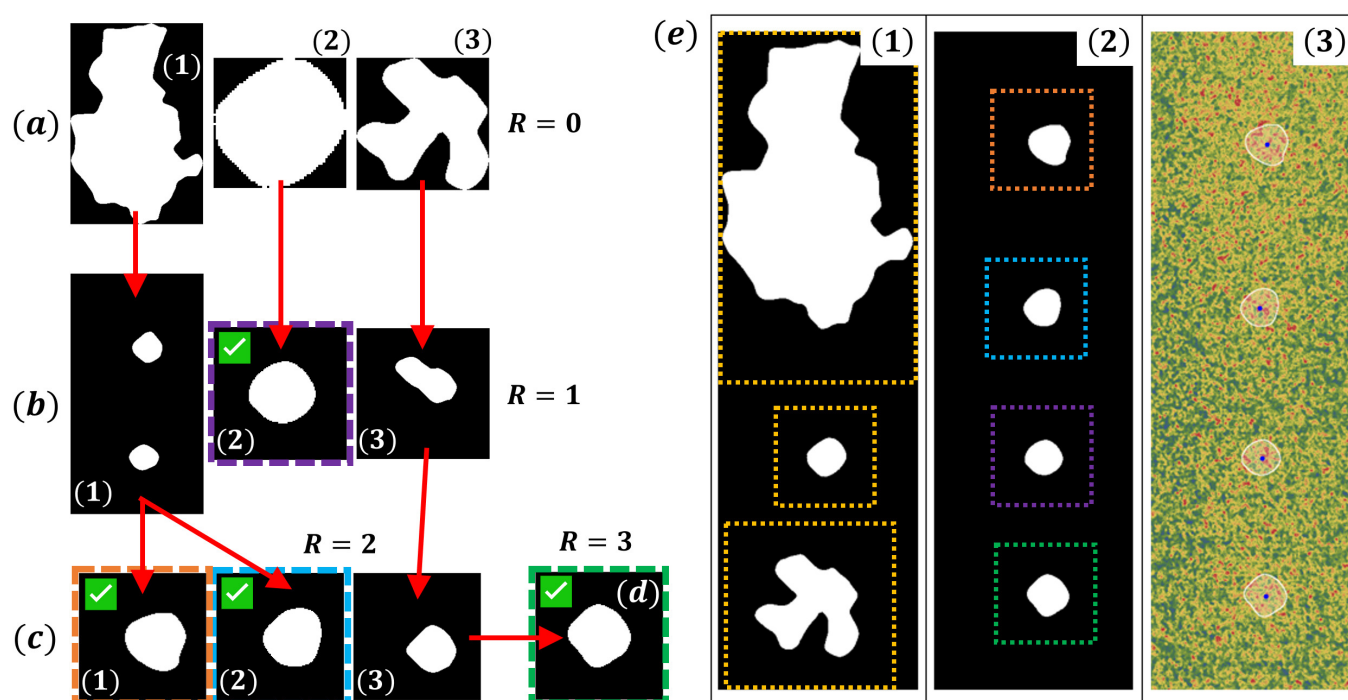

Figure 13. An illustration of the iterative interrogation process for an image with one of the worst overall SNR values, where R is the MRIF recursion depth. Sub-figures (**a**–**d**) show R running 0 through 3 for the segments detected at each depth. Sub-figure (**e**) displays (1) the initial segments, (2) MRIF output and (3) output overlays for the original image. Converged segments are flagged with green ticked boxes. Colored frames in (**e2**) correspond to (**b2**), (**c1**–**2**,**d**) via respective colors. Orange frames in (**e1**) indicate the IWs at $R = 1$: note the high aspect ratio of the topmost IW—its virtual counterpart is significantly out-of-bounds and therefore considerable crop correction (17) is assigned to properly map (**c1**–**2**) onto the FOV.

Here it is important to reiterate that the performance of MRIF strongly depends on the user-defined IW scaling factor s (8) and the critical IW length scale ratio ε for the latest and the (potential) next recursion iterations. Specifically, the $\chi = s/\varepsilon$ ratio is of interest—we suggest $\chi \geq 1$ in general. An optimal ratio, which, as we observe, should be the same for all image sequences acquired under similar conditions, enables MRIF to efficiently "strip" the first segment estimates of artefacts and perform one final update for the resolved bubbles, as in Figure 13(a3)–d. Here $\chi = 1.25$ for examples shown in Figures 11–13, with $s = 2.5$ and $\varepsilon = 2$. Before adjusting χ, we would recommend that the user determine the s value which gives the best performance for the local filter, also adjusting the settings for the latter—this will determine a starting value for ϵ before further optimization.

4.3. Results for Existing and New Data

Aside from the examples shown in Sections 4.1 and 4.2, it is also of interest how the developed approach performs for entire image sequences in terms of bubble detection density in the FOV and the physicality of obtained results, i.e., bubble trajectory and shape properties. Figures 14–16 demonstrate the differences in performance for the preceding image processing pipeline [8,43] and the methodology presented here.
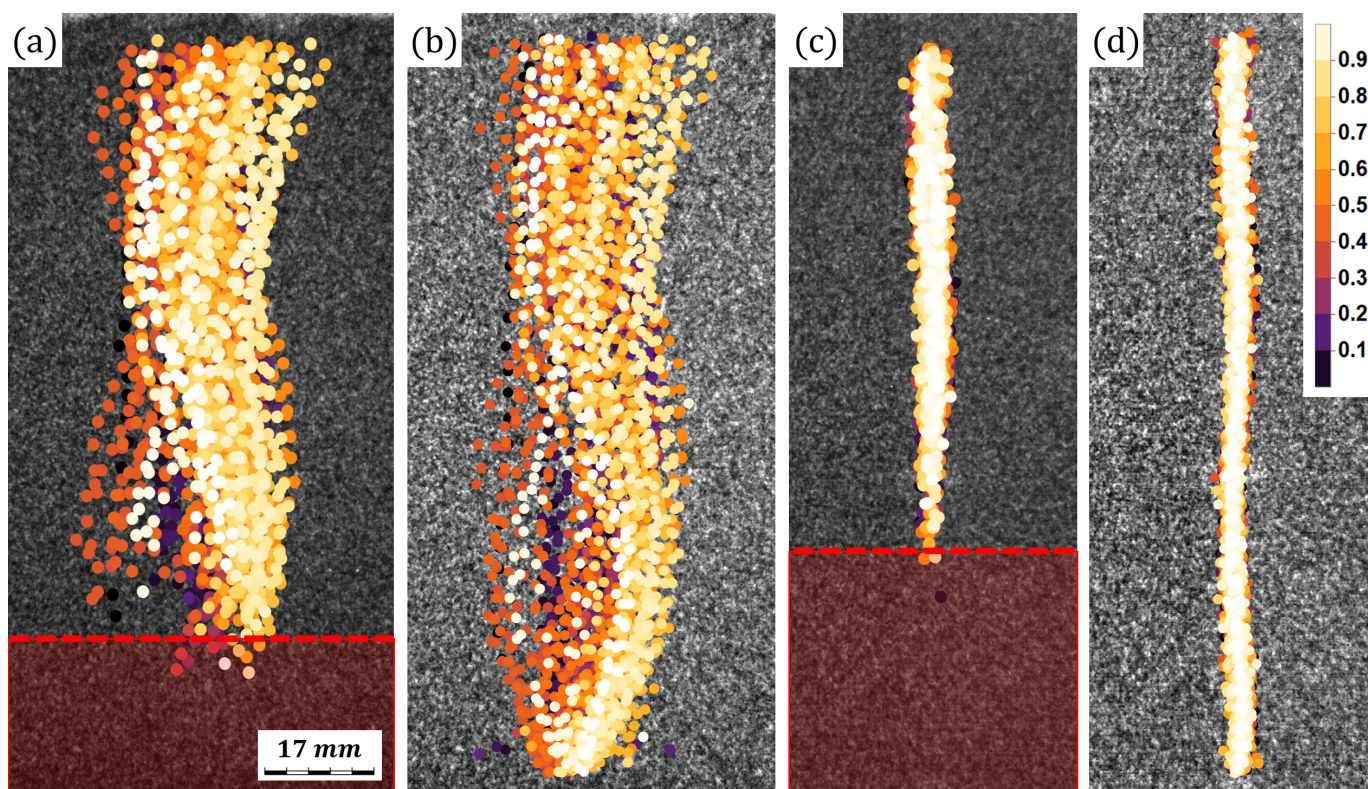

Figure 14. The locations of bubbles detected in the FOV for a sequence of 3000 images (30 s) at a 100 sccm flow rate for the model system from [8,43] (Section 2.1, horizontal inlet): no applied MF, (**a**) previous and (**b**) current image processing code; applied ∼265 mT horizontal MF, (**c**) previous and (**d**) current image processing code. Bubble locations are marked with dots color-coded in the chronological order of appearance. Note the color legend in (**d**): imaging starts at 0 and ends at 1. The red-tinted areas indicate the blind zones of the previously used image processing code. All sub-figures are to scale.
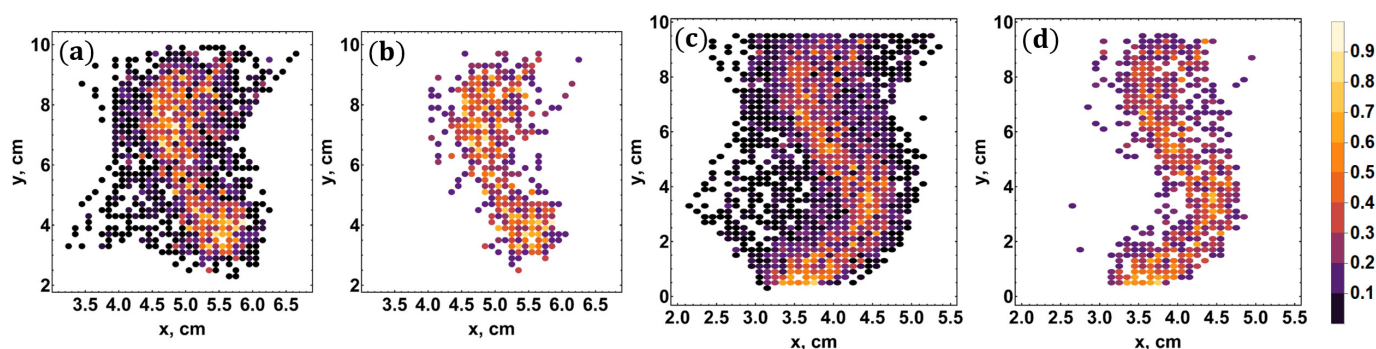
Figure 15. Normalized bubble detection density histograms with $(dx, dy) = (2, 4)$ mm bins color-coded by detection counts: (**a**,**b**) the case in Figure 14a with (**a**) all bins and (**b**) bins with 3+ detections shown; (**c**,**d**) the case in Figure 14b with (**c**) all bins and (**d**) bins with 4+ detections shown. Note the color legend to the right of (**d**).

One can clearly see in Figure 14 that the new version of the image processing code outperforms the previous version by completely avoiding the blind zones in the lower part of the FOV for both image sequences. Notice also that bubble tracks visible in (b) are much more coherent than in (a). Figure 15, in turn, shows that with the new methods one can now clearly resolve the classical S-shaped mean trajectory cluster formed by zigzag trajectories, as seen in (c,d), as opposed to (a,b) where a significant portion of the events is missing. The deflection bias in the $x > 0$ direction in (c,d) is determined by the horizontal inlet releasing gas in that direction.

Another point of interest are the tilt angle dynamics resolved in [8] versus the current results—this is showcased in Figure 16. Three things are important to note here: first, as a consequence of the blind zone elimination, the new curves extend all the way through the FOV; second, the average trends yielded by both approaches indicate that the previously used code indeed resolved the dynamics without unacceptable inaccuracy; third, the error bands are considerably narrower about the averaged curves for the present results. The latter is especially true for the case with applied MF shown in (b) where the SNR was much lower then in the image sequence corresponding to (a). This indicates that the new approach indeed yields significant improvement not just in bubble detection, but also in shape boundary resolution. The experimentally obtained results in [8] were in a rather good agreement with performed simulations, meaning one thus has an indirect yet significant validation of the presented approach.
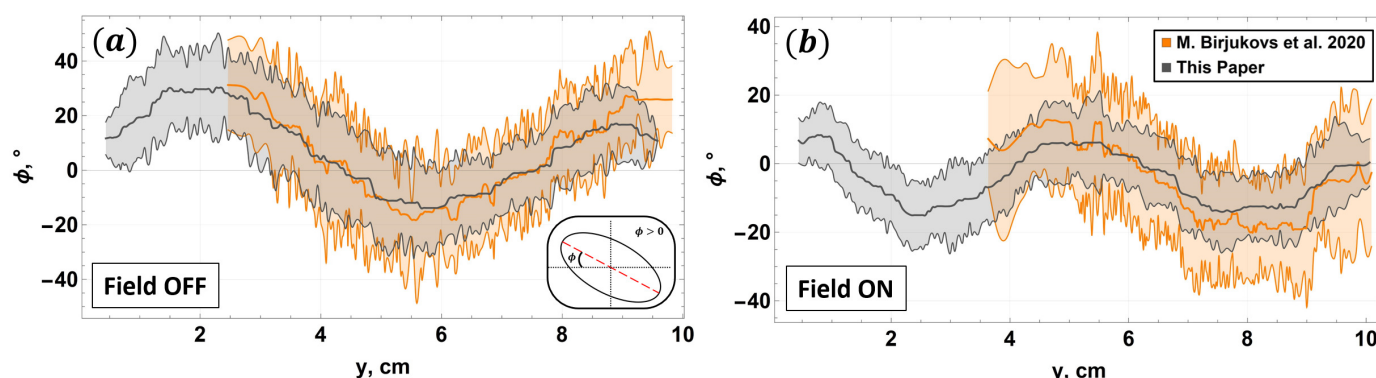

Figure 16. Bubble tilt angle versus elevation (averaged curves and error bands) over the FOV bottom for (**a**) the case with no applied MF (Figure 14a,b) and (**b**) applied horizontal ~265 mT MF (Figure 14c,d), both cases at a 100 sccm flow rate. Orange indicates the previous paper [8] and the current results are shown in gray. The tilt angle definition is shown in the bottom-right corner of (**a**).

The developed approach was also applied to the newly acquired data to ensure consistency in the code output across different experimental campaigns—one instance of

the new results is shown in Figure 17. Again, the bubbles are resolved over the entire FOV for all three cases shown. An in-depth physical analysis of the bubble dynamics is beyond the scope of this paper and is reserved for a follow-up article.

An important note on Figure 16: the averaged curves and the error bands were computed from the image processing code output as outlined in Algorithm 7. The quantile spline envelopes (QSEs) were computed following [73] and using the code (*Wolfram Mathematica* package) available on *GitHub*: Anton Antonov (*antononcube*): MathematicaForPrediction/QuantileRegression.m (accessed: 16 October 2021). For all the datasets represented in Figure 16, we used $q = 0.9$ with 3-rd order splines and $\lfloor 2.5\% \cdot N \rceil$ spline knots for QSEs where N is the number of points in the dataset; we set $N_{QSE} = \lfloor 14\% \cdot N \rceil$ and $\delta_b = \lfloor 0.5\% \cdot N \rceil$ (point density-adaptive physical bin size); the TV regularization parameters [58] were 1 for QSEs and 0.25 for the binned data.

Even though one cannot check how many bubbles the image processing code actually failed to detect without manual inspection (not feasible), one can evaluate the amount of detection events that are ruled out as false positives at the various stages of code execution for a sequence of images. The results for the five image sequences considered above (Figures 14 and 17) are presented in Table 1. Notice that in the most difficult case of the five (Figure 14d) most of the work is done by the luminance map-based filter and the object property filter. However, the intrinsic filtering capacity of MRIF is significant because it filters out the detection events that very likely would have passed both of the two following stages.
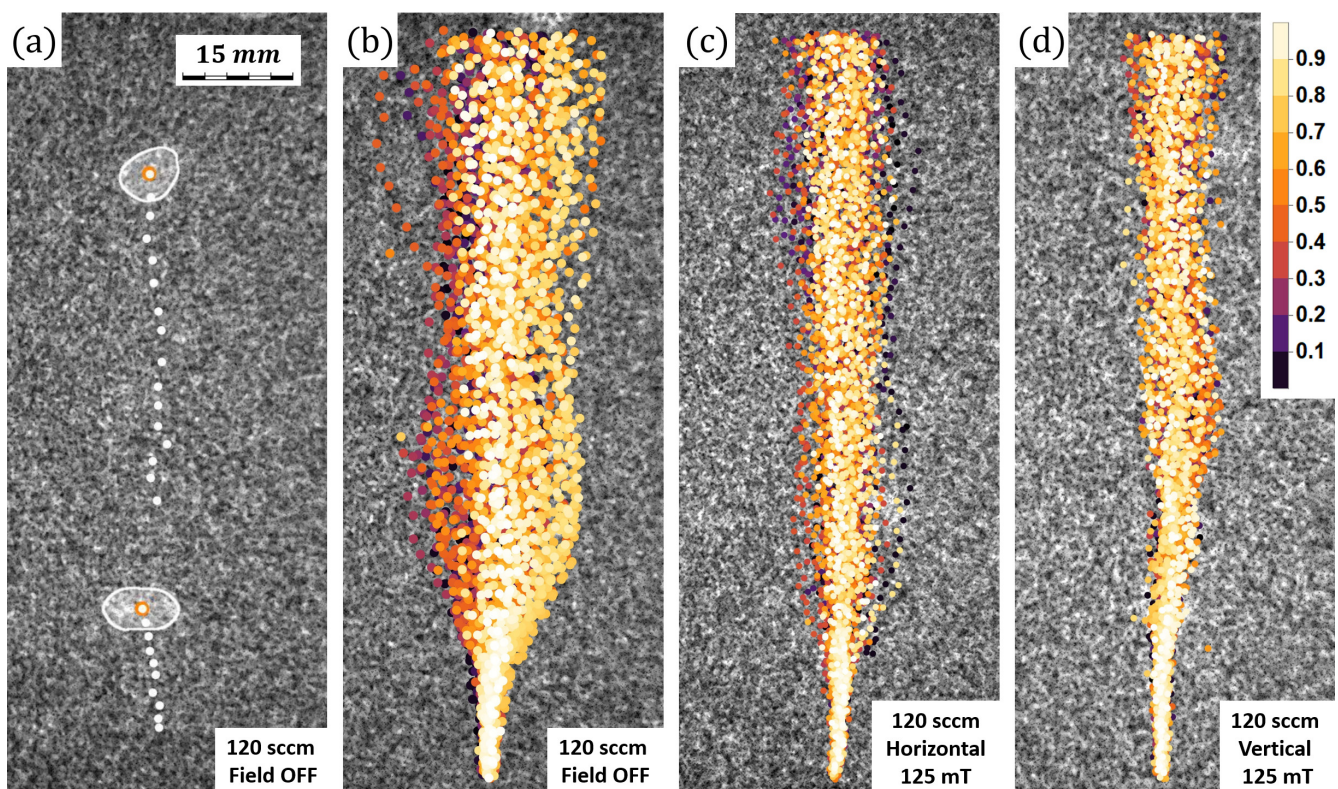

Figure 17. The locations of bubbles detected in the FOV for a sequence of 3000 images (30 s) at a 120 sccm flow rate for the new model system (Section 2.1, vertical inlet): (**a**) an example of detected bubbles: white contours are shapes, orange dots are the current positions and white dots are the preceding detections; (**b**–**d**) all detected bubble positions with (**b**) no applied MF, (**c**) horizontal ∼125 mT MF and (**d**) vertical ∼125 mT MF. Bubble locations in (**b**–**d**) are marked as in Figure 14. All sub-figures are to scale.

Table 1. False positive elimination rates (%) over three stages—MRIF, the luminance map-based filter (Algorithm 6) and the object property filter (with respect to the input that each filter received), and the percentage of detection events input to MRIF that were eliminated in total.

Image Sequence	MRIF	Algorithm 6	Property Filters	Total	Remainder
Figure 14b	1.75	2.63	1.21	5.49	94.5
Figure 14d	1.51	4.63	6.80	12.5	87.5
Figure 17b	1.53	2.05	1.25	4.76	95.2
Figure 17c	0.71	0.51	0.18	1.39	98.6
Figure 17d	1.62	0.89	0.32	2.81	97.2

Algorithm 7: Post-processing for the output of the image processing code.

Input: Tilt angle (or any other bubble property) values over elevation (or any other independent variable)

1. Construct the upper and lower quantile spline envelope (QSE) functions for the input data with the quantile q
2. Sample the QSEs evenly in intervals of N_{QSE} points over the independent variable range
3. Smooth the QSEs using Gaussian TV filtering
4. Designate the data points above the upper and below the lower lower QSEs as outliers and remove from the dataset
5. Bin the remaining data points over the independent variable range into bins with size of δ_b points
6. Compute means and standard deviations for the resulting bins
7. Smooth the resulting mean value sequence using Gaussian TV filtering

Output: An averaged curve for the dataset with error bands

---

5. Experimental Validation

5.1. Stationary Reference Body

The developed image processing algorithm is first validated by applying it to the images of a stationary reference body described in Section 2.1. Three imaging cases are considered here: neutron flux transmission through the shorter body axis, the longer axis, and the latter with an extra distance from the body to the scintillator. Thus, the SNR of the neutron-transparent spherical cavity within the body progressively decreases for these cases. This is illustrated in Figure 18.
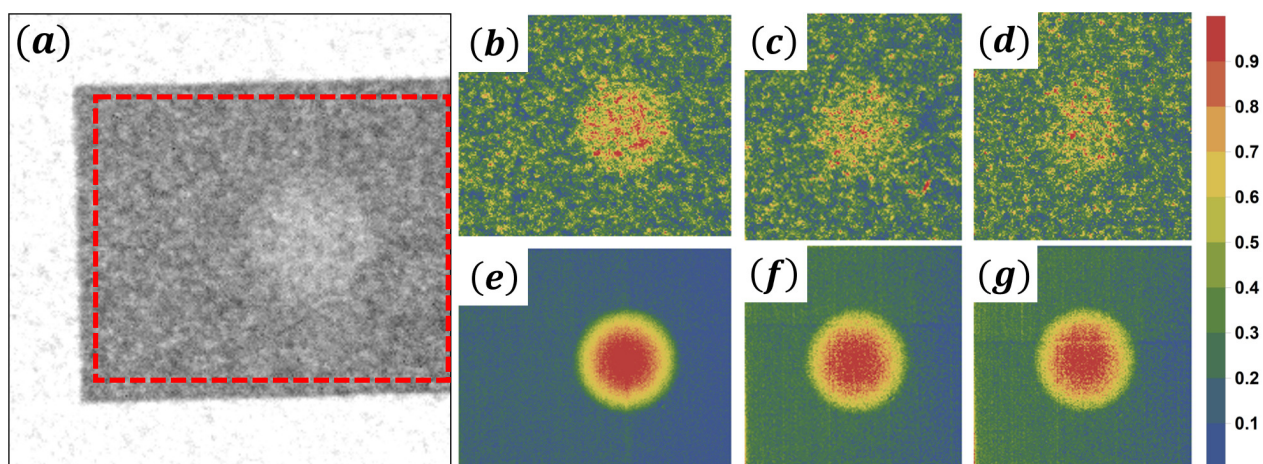

Figure 18. Static reference body: (**a**) an example radiograph; (**b**–**d**) normalized images cropped as in (**a**) for neutron flux transmission through the shorter axis, the longer axis, and the latter with an extra distance to the scintillator, respectively; (**e**–**g**) normalized mean luminance maps for the respective image sequences. Note the color bar to the right.

A neutron radiography image of the reference body (slightly inclined) is shown in (a) where one can see the rectangular brass frame (darker) and a circular projection of the spherical void (brighter), as well as the surrounding background due to air. Neutron transmission in the case of (a) is along the shorter of the body axes. In all three imaging cases the images are cropped as indicated in (a). Examples of cropped images of the spherical void within the body with exposure very similar ($\sim 1.3\times$) to that for the bubble images (100 FPS, $\sim 1.3\times$ neutron flux) are shown in (b–d) for the three cases listed above, respectively. The corresponding mean luminance maps shown in (e–g) were obtained by averaging over ~ 5.8 K, ~ 12 K and ~ 9.5 K images (entire recorded sequences), respectively— these averaged images are used to obtain reference shapes. The shapes detected by the image processing code in images like (b–d) are then compared to the reference shapes to compute shape detection error metrics. Note that all images seen in Figure 18 and used for validation are obtained from raw images by pre-processing via Algorithm 2, as with bubble flow images.

Note that the images shown in Figure 18 have the image side length to sphere diameter ratios that are very similar to what one has for MRIF IWs. To compare the reference body images to the bubble images in terms of image quality, consider Figures 19–21 versus Figure 2.
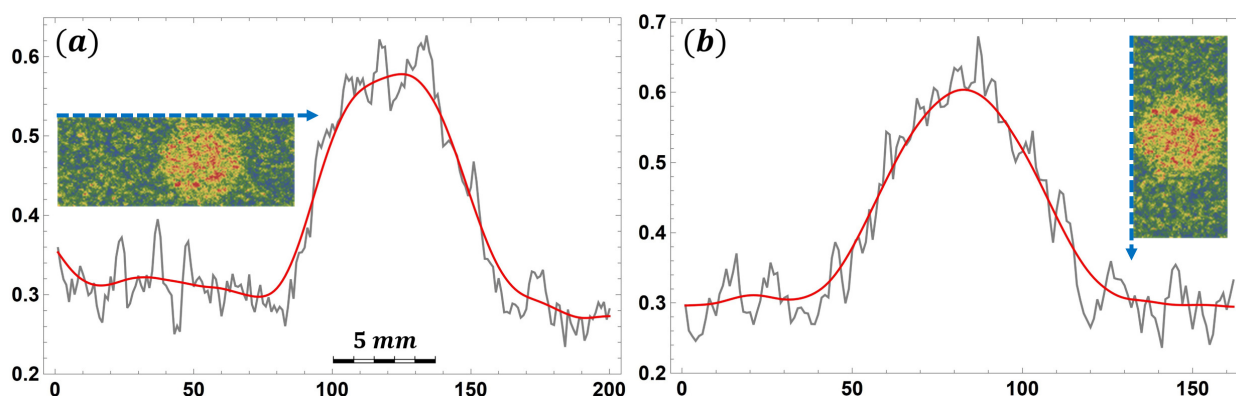

Figure 19. Neighborhood analysis for the reference spherical void, transmission along the shorter axis: normalized width mean of luminance (gray) over the length of (**a**) horizontal and (**b**) vertical patches roughly fit to the sphere dimensions, spanning the cropped body images. Note the pixel-to-mm scale bar in (**a**). Patch scan directions are indicated by the dashed blue arrows. The image patches are not to scale and their luminance maps are normalized as in Figure 18. The red curve is the total variation filtered (Gaussian, regularization parameter equal to 1 [58]) gray curve.
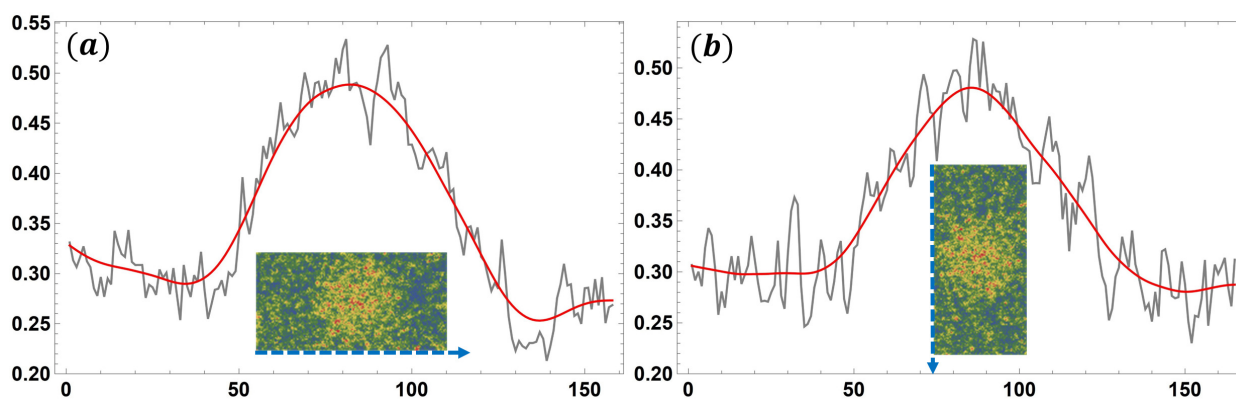

Figure 20. Neighborhood analysis for the reference spherical void, transmission along the longer axis: normalized width mean of luminance (gray) over the length of (**a**) horizontal and (**b**) vertical patches roughly fit to the sphere dimensions, spanning the cropped body images.
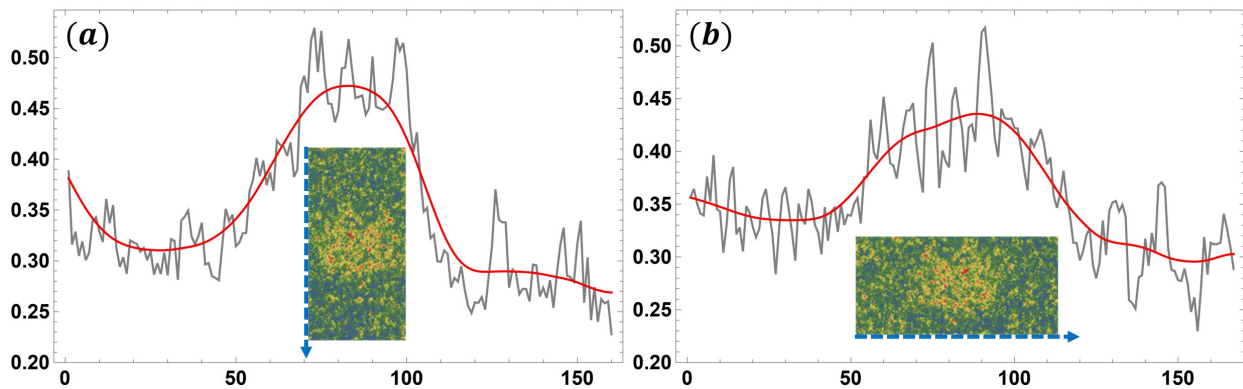
Figure 21. Neighborhood analysis for the reference spherical void, transmission along the longer axis with a extra distance to the scintillator: normalized width mean of luminance (gray) over the length of (**a**) horizontal and (**b**) vertical patches roughly fit to the sphere dimensions, spanning the cropped body images.

Notice that the case shown in Figure 21 is very similar to the bubble neighborhood case in Figure 2. In fact, Figure 21 exhibits arguably even worse CNR and SNR, meaning that, despite different materials (gallium and argon versus brass and air) and slightly different neutron flux, the reference measurements are representative of the flow imaging conditions and can be used for direct validation of the new image processing code.

To obtain the reference shapes from the images in Figure 18e–g, one first applies the Gaussian TV filter [58]

$$\frac{\partial I}{\partial t} = \nabla \left(\frac{\nabla I}{|\nabla I|} \right) + \frac{1}{\mu}(I_0 - I) \tag{18}$$

where the notation is as in (4) (note that the right-most term is different), $\mu = 0.25$ is the regularization parameter and 15 TV iterations are performed [64]. Afterwards, double-Otsu hysteresis binarization is performed followed by image border component removal and mean filtering (5-pixel radius).

To ensure fair verification, we apply both the global (Algorithm 3) and the local (Algorithm 5) filters to the reference void images with parameters identical to those used for bubble images—these are provided in Sections 3.3 and 3.4.3.

Once all images are processed and masks given by the global and the local filter are obtained, we compute the following shape detection error metrics for both filters:

- S_Δ—the area of the difference between the detected and the reference masks;
- δS—the absolute difference in the areas of the detected and the reference masks;
- δr—the absolute difference of the detected and the reference mask effective radii;
- δc—the absolute difference in the circularity (the ratio of the equivalent disk circumference to the shape polygonal length) of the of the detected and the reference mask;
- $(\delta x, \delta y)$—the absolute difference in centroid coordinates between the detected and the reference masks.

where all metrics are normalized to the respective properties of the reference masks, except $(\delta x, \delta y)$ is normalized to the reference radius.

Here the δc and δr metrics serve primarily as "red flags" against gross shape detection inaccuracies Under normal circumstances, one should have $\delta c < 3$–5% and the δr histogram should conform to that of δS. The other three metrics are used directly for shape detection error quantification where the most important one is S_Δ.

5.2. Moving Reference Body

The principles behind imaging a moving reference body are as above, except the body is now attached to a pendulum (Figure 22) that periodically swings, and thus the body travels back and forth through the FOV. The pendulum is a standard laboratory holder that

is suspended from above the neutron beam with the reference body fixed by its clamps. The motion is mostly horizontal and is initially strongly damped until the pendulum amplitude reaches a state where its oscillations exhibit a very slow decay. This enables us to determine the dynamics of the error metrics outlined in Section 5.1 and assess the effects of motion blur as the body and the void within it decelerate.

Here, before the global and local filters can be applied to the cropped reference body images, one must first segment the body within the FOV (Figure 22), crop the masked image to an IW about the mask centroid, repair the IW images as necessary, and then apply the filters. This procedure is somewhat more involved that the one in Section 5.1 and is outlined in Algorithm 8.

Algorithm 8: Void region (IW) extraction and repair from the pendulum images.

Input: A normalized pre-processed (Algorithm 2) image as in Figure 22
Image filtering
1 Poisson TV filtering (4)
2 Luminance map inversion
3 SCTMM filtering (7)
4 Image normalization
Reference body segmentation
5 Chan–Vese binarization (11)
6 Morphological opening (disk elements)
7 Morphological erosion (disk elements)
8 Delete border components
Void region extraction
9 Mask area thresholding
10 Define an IW about the body centroid based on the body segment area (8) and (9)
11 Project the input image onto the IW masking the pixels outside the body segment
Void region restoration
12 Body background extrapolation to masked IW regions using texture synthesis-based inpainting
13 Image normalization
14 (Optional) Remove high-luminance outliers and normalize the image
Output: A repaired IW containing the spherical void surrounded by the body background
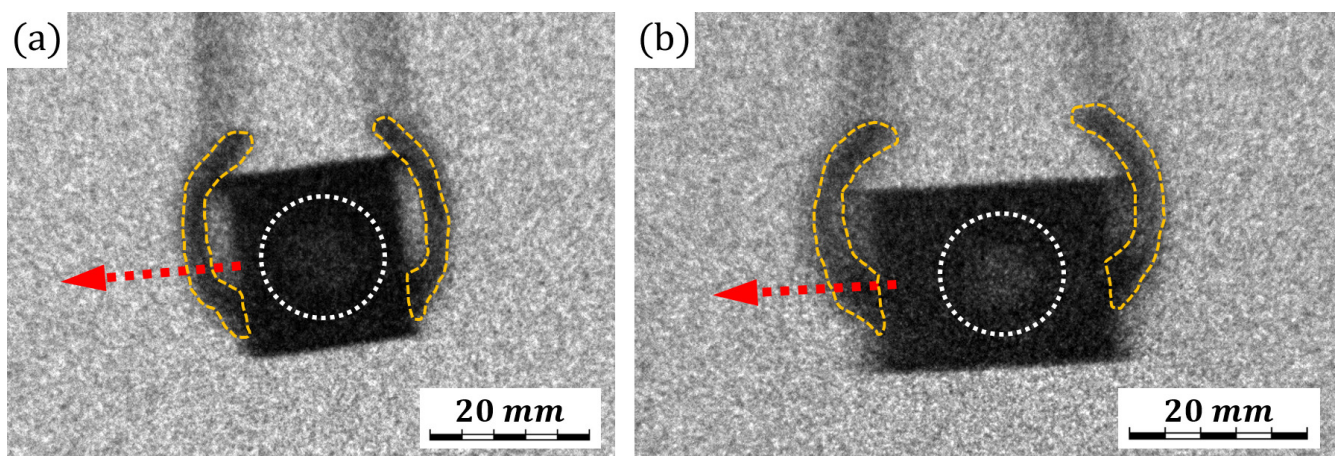

Figure 22. Neutron radiography image of the pendulum used for the reference experiments with the moving reference body: neutron flux transmission along the (**a**) longer and (**b**) shorter axes. The red dotted arrows indicate the direction of motion and the white dotted circles highlight the locations of the spherical void. The pendulum arm is a standard lab holder and the reference body is held by clamps (dashed orange lines). Note the motion blur visible at the body boundaries and the holder arms above the clamps.

The Poisson TV filter (4) reduces the noise in the image and the image luminance map inversion makes the darker body area foreground and the surrounding air (brighter) background. The SCTMM filter (7) then increases the body CNR and the body is segmented using the Chan–Vese process (11). Morphological opening (disk structural elements, 15-pixel radius) and erosion (disk structural elements, 5-pixel radius) help separate the body from the clamps (Figure 22), as well as to remove artefacts, if any. Border component removal is done because only body segments fully within the FOV are eligible for analysis. Note that here we use $c = 0.5$ for SCTMM and $\beta = 1$ for the TV filter.

After area thresholding removes mask components that have abnormally small areas (usually left over clamp segment fragments), an IW is defined about the body centroid via (8) and (9) with $s = 1.25$, and the input image with masked non-segment pixels is projected onto the IW. An example is illustrated in Figure 23. Note that (a), which shows the detected segments, contains a small artefact to the bottom left of the body segment—such segments are removed by area thresholding. An IW is then defined about the centroid of the remaining body segment and a void region is extracted, which is shown in (b). Note, however, that the IW contains portions of the masked background from (a), which can interfere with the global and local filters.

It is therefore necessary to extrapolate the body background about the void into such regions. These regions are detected by inverting the IW luminance map and running histogram-based segmentation with a single threshold of 0.999. The masked regions are then segmented from the IW, as shown in (c). Texture synthesis-based inpainting is then performed with a maximum $N_{\text{neigh}} = 150$ neighboring pixels used for texture comparison and a maximum of $N_{\text{samp}} = 300$ sampling instances for texture fitting [74].
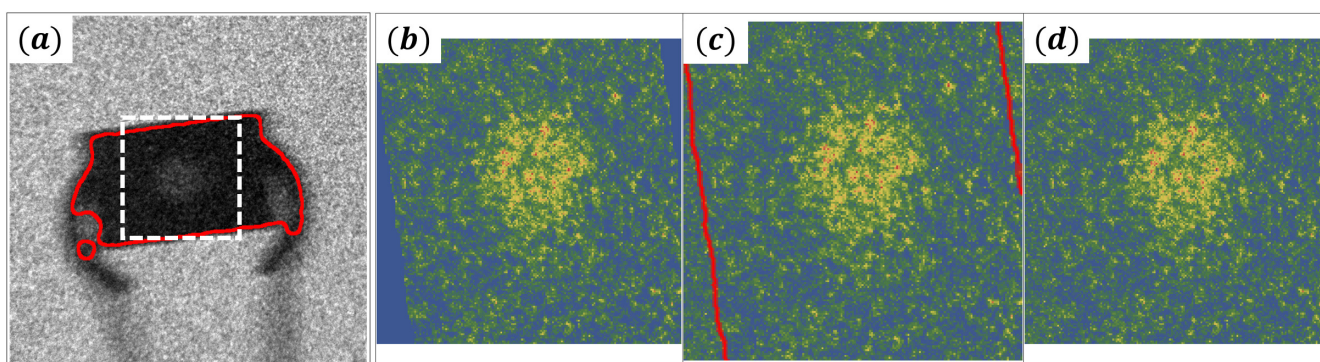

Figure 23. (**a**) Detected reference body segments (red contours) with an IW (white dashed frame) centered about the body position after segment area thresholding; (**b**) original image projected onto the IW; (**c**) image regions designated for inpainting (red borders); (**d**) restored void region.

In rather rare cases inpainting introduces pixels with strongly outlying luminance values, which are eliminated as follows. An Otsu threshold I_c is computed for the input image, and then the image is binarized using the $k_I \cdot I_c$ threshold, where k_I is the threshold scaling factor. With the right k_I value, this operation segments pixels that differ by more than a certain luminance value from the bulk of the histogram. The segmented pixels are then masked in the input image. We found that $k_I = 4$ does not affect the images without significant outliers (ones that strongly affect global and local filter output) while effectively cleaning up severe outliers without modifying the bulk image histogram. An example output of Algorithm 8 is shown in Figure 23d. Afterwards, the resulting repaired void regions are used as input for the global and local filters. The settings for the global and local filters are the same as in the cases with stationary reference bodies.

In the cases where neutron flux transmission was along the longer body axis (Figure 22a) it was more difficult to segment the body without the pendulum clamps with Algorithm 8 as outlined above. Therefore, minor adjustments were made:

- Otsu binarization was used instead of the Chan–Vese process;

- Morphological opening disk element radius was increased to 50 (Step 6);
- Morphological dilation using disk structural elements with a 5-pixel radius was performed after Step 7;
- Body masks were oriented to minimize the area of masked background in the IW before performing Step 11.

and other Steps and parameters of Algorithm 8 remained unchanged. This procedure was necessary because clamp removal from the body segments using larger-scale morphological opening resulted in body segments smaller then the body by a considerable margin, which was compensated for by morphological dilation to recover the eroded area. Body reorientation was required because the masked image margins to be filled using texture synthesis often constituted a significant fraction of the IW area, resulting in artefacts. Body orientation was detected by fitting a minimum area oriented bounding box and determining its angle ϕ (definition as in Figure 16a) with respect to the horizontal axis. The masked body image was then rotated by $-\phi$ radians and the body region was obtained and repaired as usual. In two cases we had to use lower values, $N_{neigh} = 50$ and $N_{samp} = 200$, to avoid sampling the void regions and producing high-luminance synthetic background.

In addition, filtering is performed for IWs with slightly different sizes and void positions. Therefore, one cannot directly overlay detected void shapes over a reference mask. One also cannot obtain a reference mask by averaging the images from the recorded sequence as with a stationary body. As a solution, we use the reference void segment detected from Figure 18e (best SNR and CNR) to generate a reference circle that the detected shapes will be compared against. Using the radius determined for the reference void segment, optimal circles C_{ref} are fitted to the detected segments:

$$C_{ref} = \left\{ \{\vec{c}_0, R_{ref}\} \,\middle|\, \min_{\vec{c}_0} \sum_k (\|\vec{r}_k - \vec{c}_0\| - R_{ref})^2 \right\} \tag{19}$$

where $\vec{r}_k$ are the segment boundary pixel positions, $\vec{c}_0$ is the optimized circle position and R_{ref} is the reference void radius measured from Figure 18f. This approach to reference mask placement works well only if the global and local filters are known not to systematically produce shapes with significant errors in centroid position with respect to the true void position, which has been verified using the image sequences with stationary bodies.

Given this, it is expected that (19) affects S_Δ to a degree, and δS, δr and δc are unaffected by definition. However, the centroid determination error is considerably diminished, but this is an acceptable trade-off. In fact, with (19) $(\delta x, \delta y)$ becomes a measure of circularity correlated with δc and it is arguably more useful to treat it as such. In the case of the moving reference body $(\delta x, \delta y)$ can also be used to quantify shape detection errors induced by motion blur.

5.3. Error Analysis

With the images from the reference experiments processed and the error metrics calculated, one can now assess the performance of the developed code more strictly than in Section 4.3. Starting with the imaging for a static reference body, the results are presented in Figures 24–26.

One can see in the S_Δ histograms (Figure 24) that the global filter error distribution peaks and means are shifted towards higher S_Δ values going from neutron flux transmission along the shorter (a) to longer (b) axis, and value dispersion is also increased with considerably more instances of $S_\Delta > 10\%$. When the body is moved 1 cm away from the scintillator (c), the global filter peak is shifted further towards higher values but the dispersion does not change significantly. The maximum values (not shown) are higher as well with respect to (b). Similar tendencies can be observed for the local filter errors, but the overall error values are considerably decreased and distribution peaks are shifted back to lower S_Δ values. The S_Δ dispersion and maximum values are also diminished across the board. Thus, while not radical, the improvements due to local filtering are still very clear.
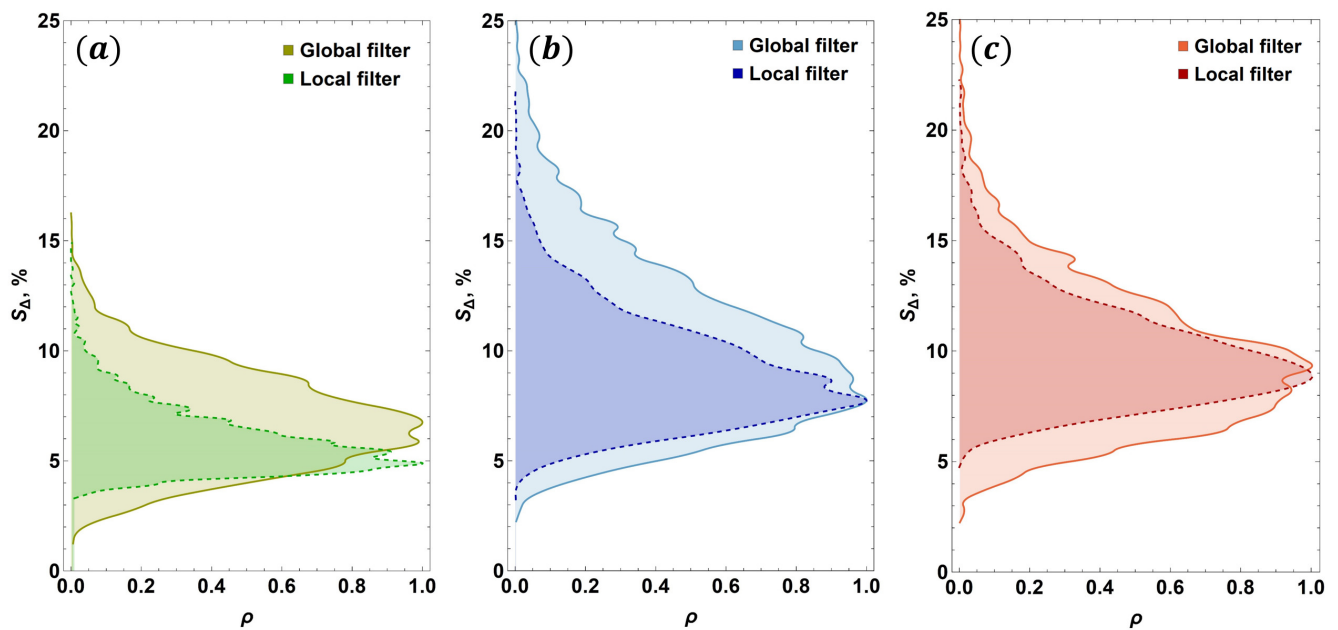
Figure 24. Static reference body: smooth normalized S_Δ histograms for neutron flux transmission along the (**a**) smaller (20 mm) axis, (**b**) larger (30 mm) axis and (**c**) larger axis with an extra 1 cm distance between the scintillator and the body (0 mm by default). Note the color legends in the upper right corners of (**a**–**c**) indicating results for the global and local filters. Here ρ is the normalized event density. Histogram bins were determined using the Scott method and 2-order interpolation was applied to bin density values.
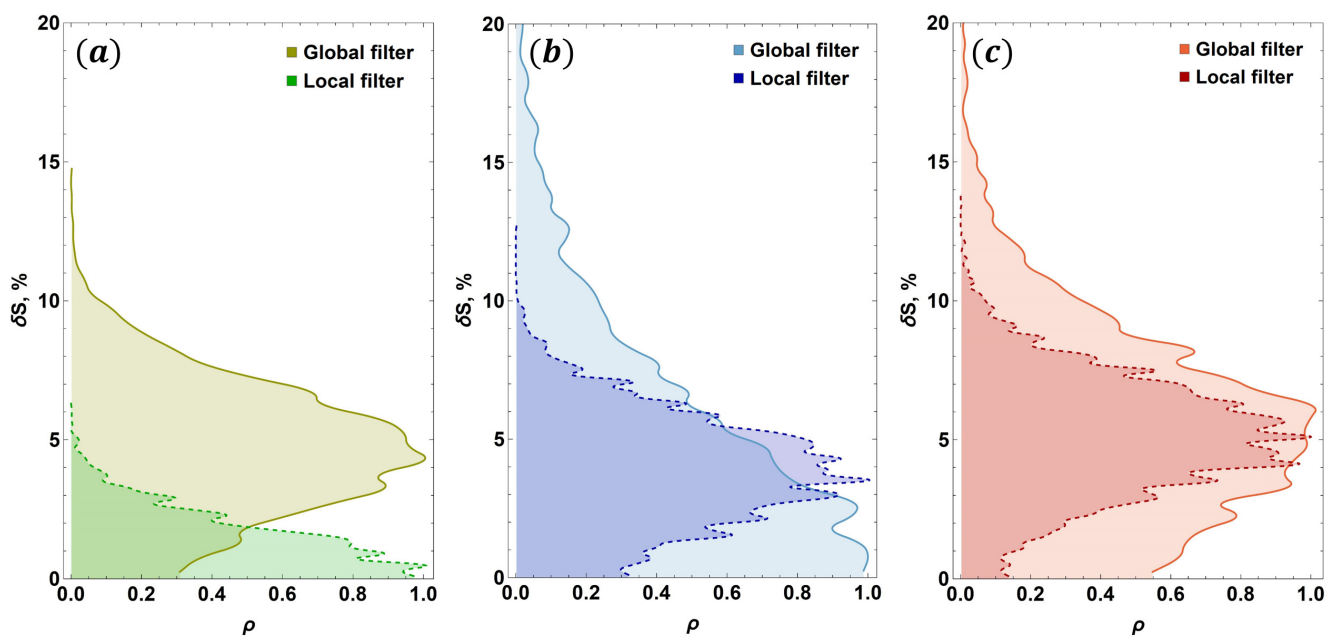

Figure 25. Static reference body: smooth normalized δS histograms for neutron flux transmission along the (**a**) smaller axis, (**b**) larger axis and (**c**) larger axis with an extra 1 cm distance between the scintillator and the body.
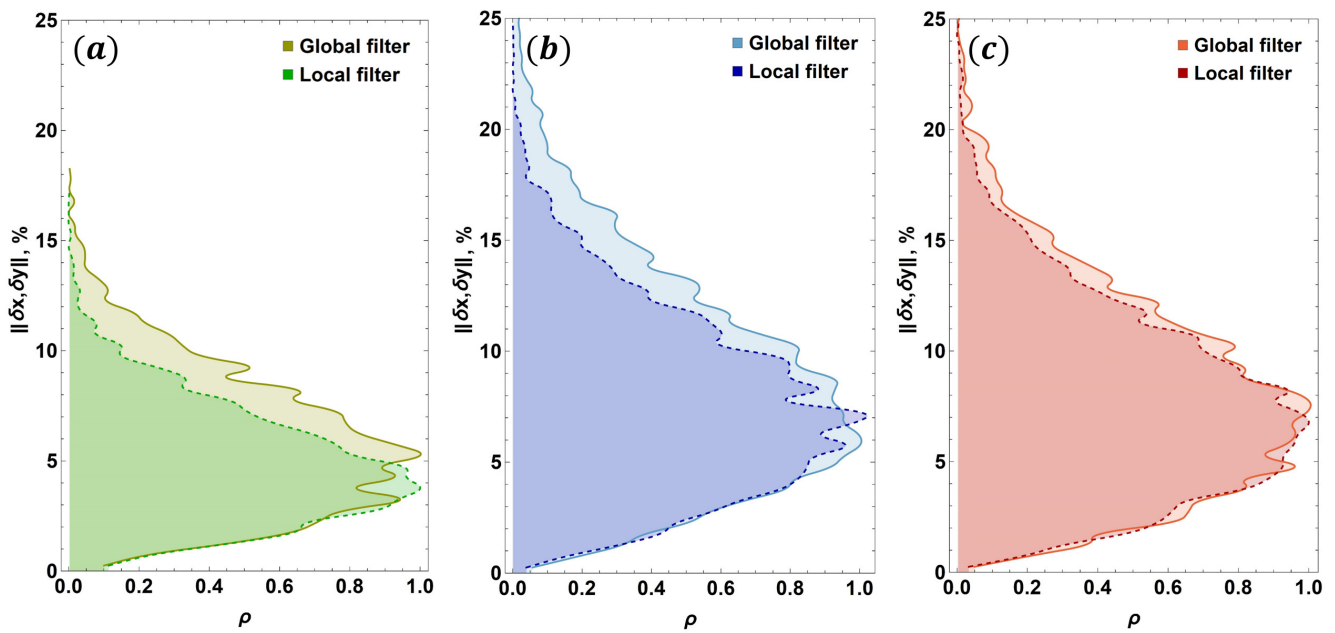
Figure 26. Static reference body: smooth normalized $\|\delta x, \delta y\|$ histograms for neutron flux transmission along the (**a**) smaller axis, (**b**) larger axis and (**c**) larger axis with an extra 1 cm distance between the scintillator and the body. Note that here $\delta x \approx \delta y$ for all instances, which is why they are not plotted individually.

The δS distributions shown in Figure 25 are different in that there are many more instances of $\delta S < 5\%$. Note that, while in (a) where there are relatively less $\delta S < 5\%$ values than in (b,c) for the global filter, the local filter improves the results significantly—dispersion is minimized and the distribution mean is shifted below $\delta S = 2\%$. This is in contrast to what is seen in (b,c): in (b), the δS peak is shifted towards larger values by the local filter, and in (c) it stays at roughly the same position. However, again, the local filter drastically minimized δS dispersion and maxima. The reason why one observes relatively less $\delta S < 5\%$ values in (b,c) than in (a) is that the local filter performance depends on how well the first three stages of the global filter improve SNR and CNR. That is to say, most of the instances with $\delta S \gtrsim 7\%$ are converted to values about the peaks of the local filter error distribution because the local filter in most cases cannot achieve improvement to below than $\delta S \lesssim 3\%$. Therefore it stands to reason that less such values are seen in (c) than in (b). As with S_Δ (Figure 24), one can see that the local filter performs progressively worse from (a) to (c), as expected.

In the static body case one would expect $\delta x \approx \delta y$, which is indeed the case, since there is no bias due to motion blur. Figure 26 therefore shows the norm of the position error vector. Notice that here the local filter improves the error distributions only sightly, reducing the relative amount of higher $\|\delta x, \delta y\|$ values in all cases. It is important to point out that δr distributions conform to δS plots and the δr metric values are lower by a factor of 2, while δc is negligible across the board. The error values seen in Figures 24–26 are within acceptable ranges.

Moving on to the moving reference body imaging, the results of these tests are shown in Figures 27–32. Figure 27 shows the pendulum (and reference void) velocity dynamics over consecutive frames for all imaging series. The [20; 40] cm/s range of expected bubble velocities (Section 2.3) is covered by the performed measurements, as seen in (a) and (a1), where the obtained velocity range is [5; 70] cm/s. Importantly, in addition to shorter and longer axis transmission tests, we also used purposefully inappropriate texture synthesis and outlier removal settings in Algorithm 8 to generate three sets of data with synthetic image edge artefacts (luminance similar to the void regions) and single pixels with luminance exceeding the image maximum by an order of magnitude (greatly

reduced image CNR) to see how the filters perform. Consider Figure 28 where S_Δ and δS distributions are shown for the three test groups.

As before, the global filter performance in terms of both S_Δ and δS progressively decreases as the transmission axis length is increased and then image artefacts are introduced to the data from the longer axis transmission measurements. Notice that the S_Δ and δS values in Figure 28 are in all cases greater than in the static body cases (Figures 24–26). This is clear enough to be seen visually, as histogram peaks in (a,b) are just under $\lesssim$ 15% errors for the global filter output, while only being just under $\lesssim$8% in the worst case for S_Δ in Figure 24c. However, this is exactly where the local filter makes a radical difference—observe in Figure 28a,c,e that the dispersion, maxima, minima and means of S_Δ are greatly reduced. Similarly good performance is seen in (b,d,f) in terms of δS. Notice that the changes in the local filter error distributions as image quality gets worse from (a,b) to (e,f) are consistent with what is seen for static body imaging in Figures 24 and 25. Another important point here is that two of the shorter axis transmission instances (Figure 28a,b) and one from the longer axis groups (Figure 28c,d) are with an extra 1 cm body-to-scintillator distance (2 mm by default)—however, in this case the differences are not significant enough to warrant attention. This likely stems from the fact a large fraction of errors is due to motion blur, especially during the pendulum deceleration stage, which is within the first 300–500 imaging frames (Figure 27). It is also very clear from Figure 28e,f that introducing additional image artefacts considerably degrades the global filter performance. The local filter, again, greatly improves the quality of detected shapes, but, of course, also produces greater errors as opposed to what is seen in (c) and (d).
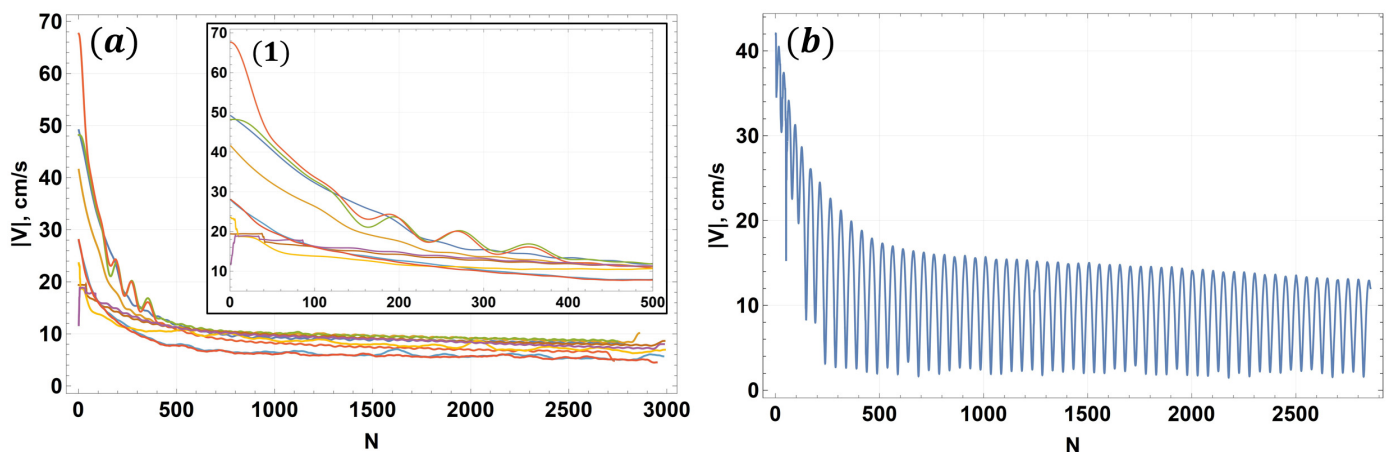

Figure 27. Frame pair velocimetry (magnitude) for the moving body (Figure 22): (**a**) velocity over sequential frames for all recorded image sequences with oscillations filtered out for visual clarity; (**b**) velocity for one of the image sequences with oscillations shown. (**a1**) shows the first 500 frames in (**a**). Gaussian TV filtering was used for (**a,b**) with the regularization parameters set to 5 (150 iterations) and 0.5, respectively.

One must remember that S_Δ is reduced by (19). To estimate the reduction, consider that if the centers of two circles with equal radii are displaced by a factor $k \leq 2$ of the radius of either, the relative area of their difference is $S_\Delta(k) = 1 - (2/\pi)\arccos(k/2) + (k/\pi)\sqrt{1 - k^2/4}$. Assuming the worst-case scenario where mask fitting via (19) "improves" the position of the detected shape by the peak $\|\delta x, \delta y\|$ value in Figure 26c, $k \sim 8\%$, the S_Δ underestimation comes out to $S_\Delta^+ \sim 5\%$. To verify this, we also observe how S_Δ changes for shapes with significant deformities detected (rarely) for stationary body images when ground truth masks (Figure 18e–f) are replaced with fits using (19)—this is in agreement with the above idealized estimate rather well, yielding S_Δ differences $\in (4; 5\%)$. If this adjustment is applied to Figure 28a,c,e, the observations are consistent with the results for the stationary reference body, and the local filter S_Δ values come out to about $\sim 10\%$ at distribution peaks (versus $\sim$8% for Figure 24c), excluding one of the cases in Figure 28e (light green curves). To reiterate, δS, δr and δc are unaffected by (19) by their definitions.

Here the δr distributions again conform to δS. δc is negligible for all cases and, as seen in Figure 29, $\|\delta x, \delta y\|$ is such that the asphericity of the detected shapes in within acceptable bounds.

Turning to Figure 29 and recalling that with (19) δx and δy are correlated with the sphericity of the detected shapes, notice that the errors are anisotropic in all cases. The greater component δx corresponds to the main direction of motion. Importantly, this anisotropy is also observed when inspecting the shape/reference difference masks for the local and global filters, where the largest contributions to S_Δ are at the shape sides in the $-x$ and x directions which are the pendulum (Figure 22) oscillation directions.
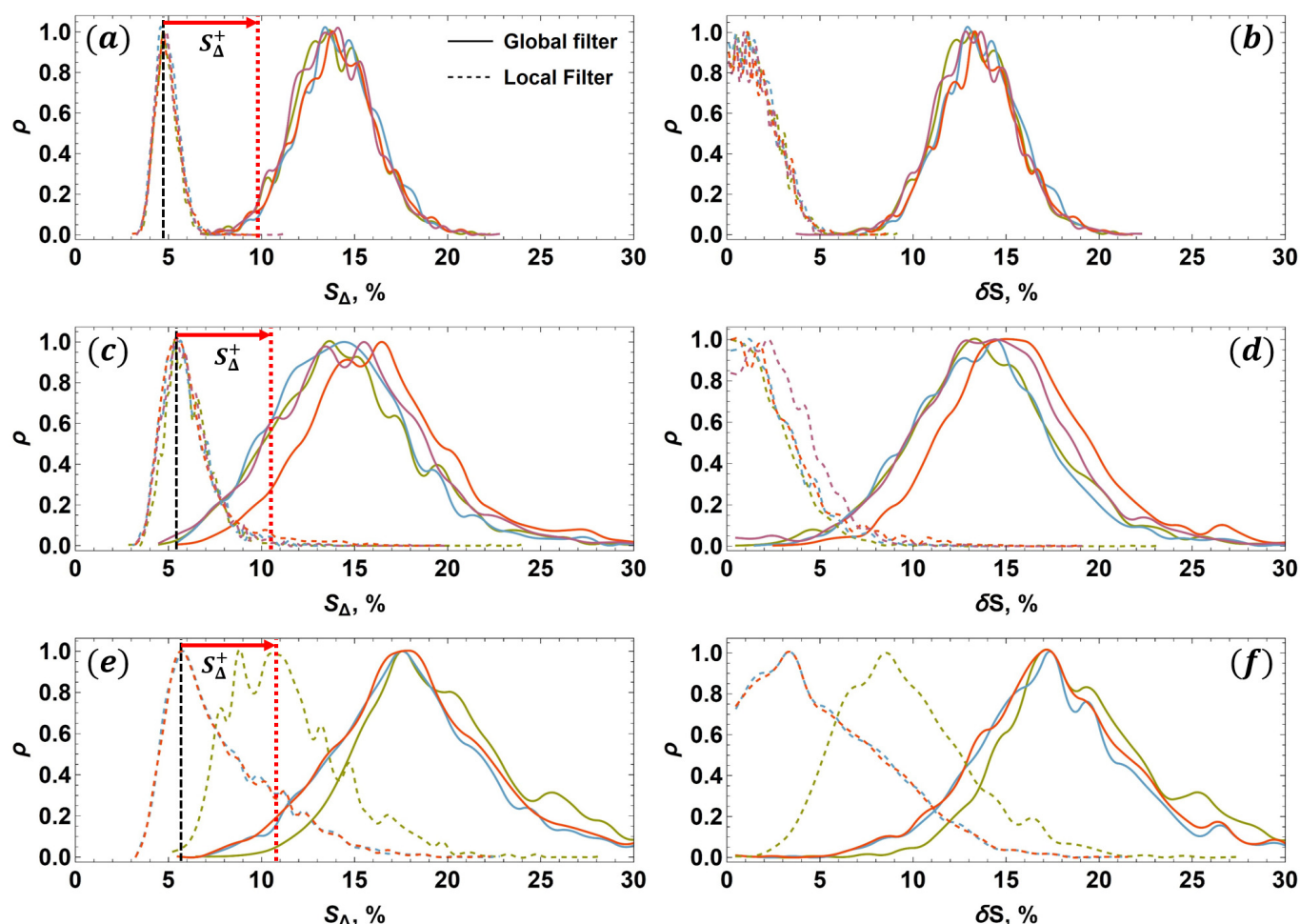

Figure 28. Smooth normalized S_Δ (**a,c,e**) and δS (**b,d,f**) histograms for the cases with a moving body (pendulum): neutron flux transmission along the (**a,b**) smaller axis, (**c,d**) larger axis and (**e,f**) the latter with the addition of synthetic image artefacts. Note the legend in the upper-right corner of (**a**) indicating the results for the global and local filtering. Colors indicate different imaging instances. S_Δ^+ is the correction for the reference mask fitting bias (red arrows).

Since S_Δ and δS maxima for the global filters were not included in Figure 28 for visual clarity, and to quantify error dispersion, Figures 30 and 31 show these quantities explicitly for all three test groups and for both global and local filters, while shape detections with S_Δ and δS values close to the values indicated in Figures 30b and 31b are very rare, it is important that the local filter can significantly improve the quality of detected shapes even in these instances.
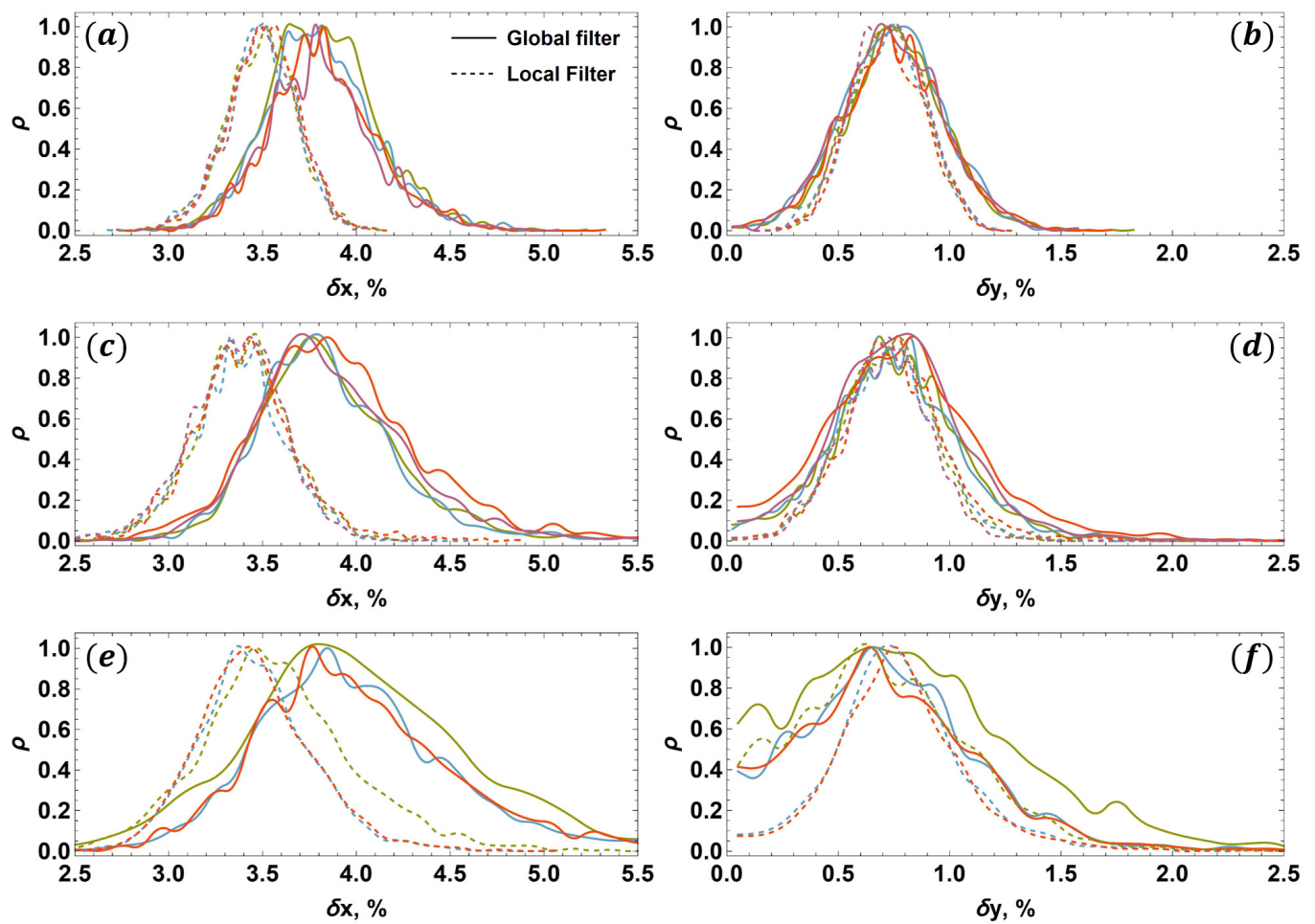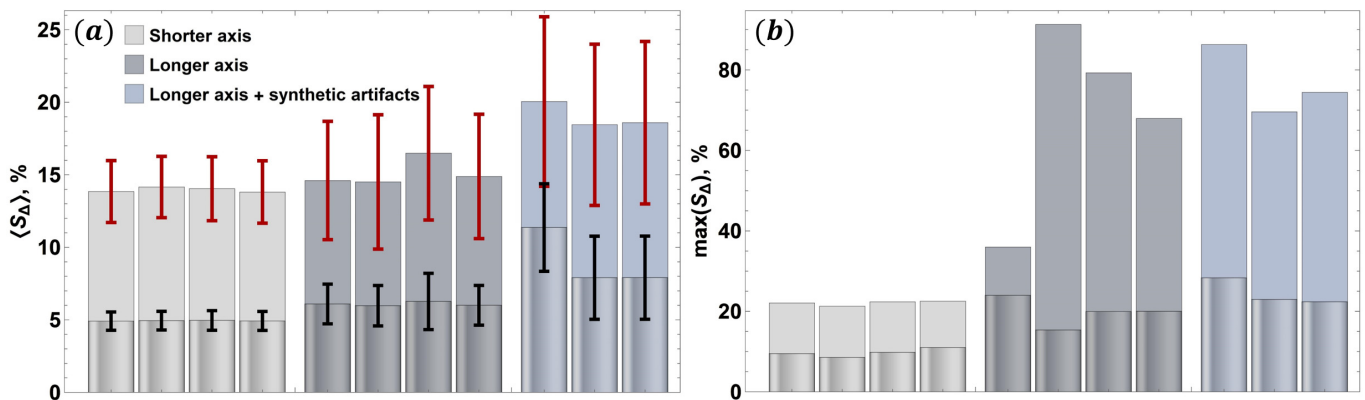
Figure 29. Smooth normalized δx (**a**,**c**,**e**) and δy (**b**,**d**,**f**) histograms for the cases with the moving body: neutron flux transmission along the (**a**,**b**) smaller axis, (**c**,**d**) larger axis and (**e**,**f**) the latter with the addition of synthetic image artefacts.

Figure 30. (**a**) Mean and (**b**) maximum S_Δ values for the cases with a moving reference body. The results for the global and local filtering are represented with matte and glossy bars, respectively. The standard deviations for the mean values in (**a**) are indicated by the red (global) and black (local) error bars. Note the color legend in the upper-left corner of (**a**) indicating the three different test groups considered in Figures 28 and 29.
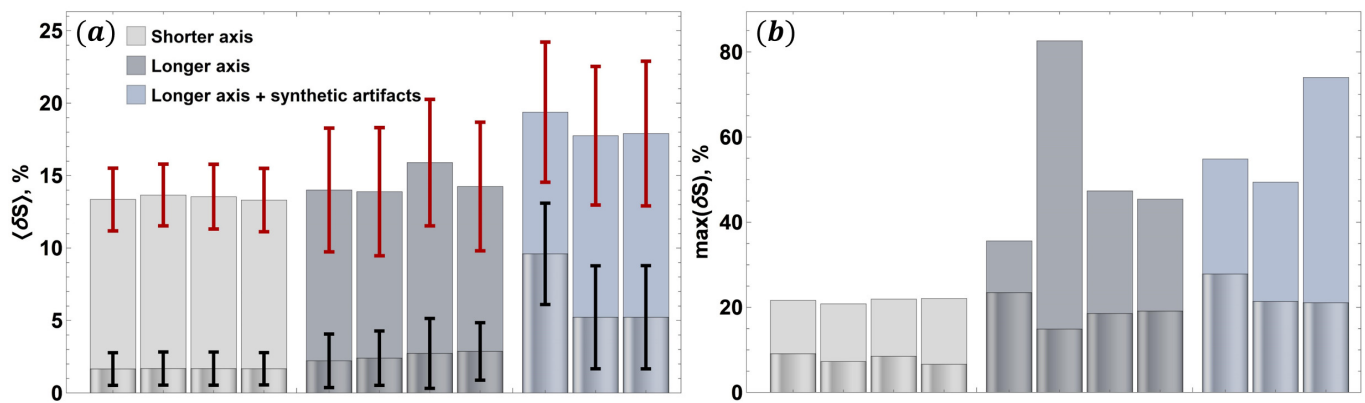
Figure 31. (**a**) Mean and (**b**) maximum δS values for the cases with a moving reference body. The results for the global and local filtering are represented with matte and glossy bars, respectively.

The effects of motion blur can be assessed more quantitatively and directly—consider Figure 32 where S_Δ is plotted as a time series for two of the processed image sequences. One can see that higher body motion velocity indeed corresponds to greater errors, which decline over time as the body decelerates. Note that, between Figure 32a,b, it is evident that the intrinsic error signal due to the global filter quickly obscures the error contribution due to motion blur. However, it also seems that the global filter is affected by the motion blur more than the local filter—as seen in (b), the S_Δ rather quickly relaxes to a quasi-stationary value at about $N \sim 250$, while in (a) the error decay persists until $N \sim 500$. Similar dynamics can be observed in (c) and (d) as well.
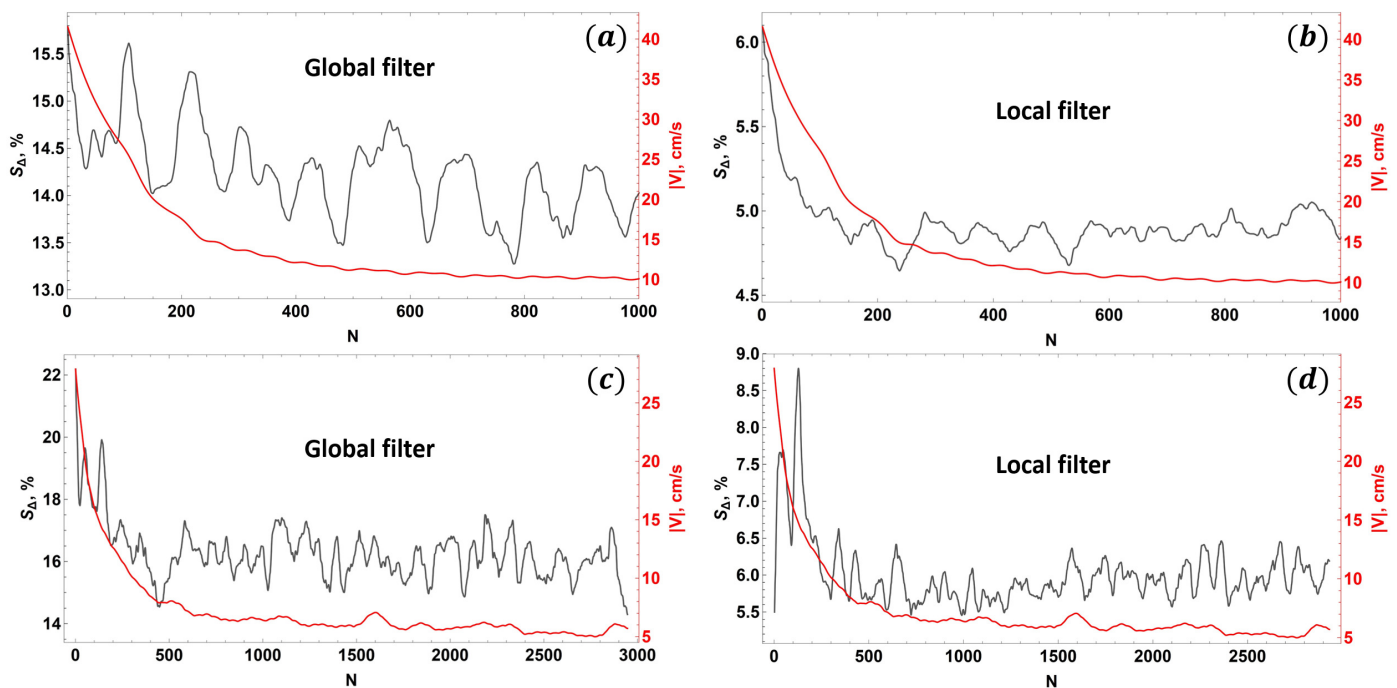

Figure 32. An illustration of the effect of body motion on the shape detection errors: examples from two image sequences. Neutron transmission along (**a**,**b**) the shorter and (**c**,**d**) longer axis of the reference body. (**a**,**c**) show the S_Δ dynamics over consecutive frames for the global filter and (**b**,**d**) show the results for the local filter. (**a**,**b**) show the first 1000 frames for visual clarity. Note that 3000 images were analyzed in total for both shown cases. The error time series show here are obtained from raw data by removing the outliers above and below the Gaussian TV-filtered (regularization parameter 1) $q = 0.9$ QSEs (3-rd order splines, $\lfloor 90\% \cdot N \rceil$ spline knots) and applying the Gaussian TV filter (regularization parameter 2) to the remaining data points. The velocity curves are as in Figure 27a.

We assess the reference void detection failure rates in the image sequences used for the code validation—the results are summarized in Table 2. There are no detection failures for static body imaging and for a moving body with neutron flux transmission along the shorter axis, and very small percentages are found for the longer axis case. The cases with synthetic artefacts have significantly higher failure rates for the global filter. However, all cases exhibit values < 4%, which indicates acceptable performance.

Table 2. Reference void failure rates (%) for the validation experiments. Static: tests (1–3) are with neutron flux transmission along the shorter axis (SA) of the body, the longer axis (LA), and the latter with an extra distance to the scintillator. For moving body imaging, tests with an extra body-to-scintillator distance are (3–4) for SA and (4) for LA. Tests with LA and synthetic artefacts are based on data from LA runs (2) and (4).

Filter	Static			Motion: SA				Motion: LA				Motion: LA + Artefacts		
	1	2	3	1	2	3	4	1	2	3	4	1	2	3
Global	0	0	0	0	0	0	0	0.33	0.37	1.24	1.10	3.08	2.51	2.85
Local	0	0	0	0	0	0	0	0.44	2.54	1.71	1.27	0.87	1.83	1.80

Finally, Table 3 indicates peak errors induced by motion blur at near the maximum pendulum velocity for the local filter for all the imaging instances, including $S_\Delta^+ \sim 5\%$.

Table 3. Peak S_Δ values (including S_Δ^+) for the local filter at near maximum body velocity for all validation experiments.

	Motion: SA				Motion: LA				Motion: LA + Artefacts		
	1	2	3	4	1	2	3	4	1	2	3
max $\lvert V \rvert$, cm/s	49.1	41.5	48.1	50.2	19.0	19.2	27.3	28.0	19.2	28.0	28.0
max S_Δ, %	11.5	11.1	11.1	10.8	12.2	13.0	13.8	13.2	19.6	18.4	18.5

Note that the peak errors are within 14% for tests without synthetic artefacts, and within 20% for instances with artefacts added to images.

6. Further Improvements and Extensions

While the demonstrated image processing performance is satisfactory for the problems that the code was developed for, its degree of parallelization could still be increased, especially for MRIF, and we expect that significant speedup can be attained for several components.

The developed code has been tested using the following hardware:

1. Intel Core i7-7700 (accessed: 16 October 2021) (4 cores/8 threads) with 64 Gb 2400 MHz DDR4 RAM;
2. Intel Core i7-8700 (accessed: 16 October 2021) (6 cores/12 threads) with 64 Gb 2400 MHz DDR4 RAM;
3. Intel Core i9-9900K (accessed: 16 October 2021) (8 cores/16 threads) with 64 Gb 2666 MHz DDR4 RAM;
4. Intel Xeon W-2255 (accessed: 16 October 2021) (10 cores/20 threads) with 192 Gb 2933 MHz DDR4 ECC RAM;
5. Intel Core i9-10980XE (accessed: 16 October 2021) (18 cores/36 threads) with 256 Gb 2933 MHz DDR4 RAM.

and we found that memory utilization for parallel execution of Algorithm 3 using hyper-threading (all images are processed independently) and all available threads for a sequence of 3000 images (properties given in Section 2.2) requires almost all of the memory for the first and the last two machines, while the second and third machines ran out of memory and the image batch size had to be decreased, while this is not a critical issue, the mean execution time per 1000 images reduces significantly between machines as the thread count increases, so we expect the reduction in memory utilization to be worthwhile. For context,

the first machine fully processes 3 K images and outputs results in ∼2 h, while the fourth machine finishes in ∼1.3 h. A more detailed performance report will be provided in a follow-up publication for a greater number of processed image sequences.

It is also planned to implement a feedback loop that will enable coupling with our recently developed object tracking algorithm MHT-X [75] for iterative reinforced object detection and tracking. Finally, we will also apply the developed methods to image sequences with even smaller bubble-bubble distances where bubble collisions also occur.

7. Conclusions

To summarize, we have demonstrated the new version of our image processing methodology for resolving gas bubbles travelling through liquid metal imaged using dynamic neutron radiography. The showcased components of our code, such as the multi-scale recursive interrogation filter (MRIF) and the underlying global and local image filters, as well as soft color tone map masking (SCTMM), proved effective for detecting bubbles and extracting their dynamic shapes from images with low SNR and CNR. Output quality was further improved by the implemented luminance map-based false positive filter that bolstered the MRIF's intrinsic false positive filtering function.

It as shown by direct comparison that the new image processing code clearly outperforms the previous version used in [8,43], while the outputs of both are still consistent. In addition, we have validated the new methods experimentally by imaging a reference body, both stationary and in motion, with a precisely machined spherical cavity. Results indicate that that local filtering performed by MRIF largely limits the shape detection errors: relative shape mismatch area and shape area difference with respect to reference shapes are within acceptable bounds of ∼14% (∼20% with synthetic artefacts) and ∼10% (∼14% with synthetic artefacts), respectively, (accounting for motion blur and the worst-case underestimation correction), while the asphericity of the detected shapes is rather negligible. As such, we find that applying the current methodology to the neutron radiography images obtained for our model systems with bubble chain flow is safe in that physically meaningful results with manageable errors can be expected. Note that we have also used the present image processing code to benchmark our object tracking code MHT-X [75].

In follow-up articles, we are going to process the data acquired in the previous and latest neutron imaging campaigns using the methods presented in this paper and our MHT-X code, and showcase the effects of applied horizontal and vertical magnetic field with different strengths on bubble chain flow in a rectangular liquid gallium vessel. Bubble trajectories (length, curvature, oscillation frequencies, envelopes, etc.), velocity (both overall spectra and dynamics, including acceleration) and shapes (aspect ratio, tilt angle, etc., and dynamics thereof) will be assessed and compared for different flow conditions—in addition to magnetic field configurations, a range of gas flow rates will be considered. We will also attempt to perform dynamic mode decomposition for the bubble shapes extracted from neutron radiography images and compare the dynamics against simulations—this will be done using the methods recently developed in [55].

Finally, we expect that the developed image processing pipeline and/or separate elements thereof should be applicable beyond the current application and context, which we also plan to demonstrate in follow-up papers. This is especially true for MRIF which is modular in the sense that the global and local filters used herein could be substituted by any other methods (both simpler ones and more complex state-of-art) depending on the problem at hand. The image processing code is available on *GitHub*: Mihails-Birjukovs/Low_C-SNR_Bubble_Detection (accessed: 16 October 2021). The code will be improved as outlined in Section 6.

Author Contributions: Conceptualization, M.B.; data curation, M.B., P.T. and A.K.; formal analysis, M.B.; funding acquisition, K.T., A.J. and M.B.; investigation, M.B., P.T., A.K., J.H. and M.K.; methodology: M.B., P.T. and M.K.; project administration, M.B. and A.J.; resources: P.T., A.K., J.H. and D.J.G.; software: M.B.; supervision: K.T. and A.J.; validation, M.B.; visualization, M.B.; writing—original

References

1. Baake, E.; Fehling, T.; Musaeva, D.; Steinberg, T. Neutron radiography for visualization of liquid metal processes: Bubbly flow for CO_2 free production of Hydrogen and solidification processes in EM field. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Dresden, Germany, 19–20 September 2017; Volume 228, p. 012026. [CrossRef]
2. Liu, Z.; Li, L.; Qi, F.; Li, B.; Jiang, M.F.; Tsukihashi, F. Population Balance Modeling of Polydispersed Bubbly Flow in Continuous-Casting Using Multiple-Size-Group Approach. *Metall. Mater. Trans. B* **2015**, *46*, 406–420. [CrossRef]
3. Schurmann, D.; Glavinic, I.; Willers, B.; Timmel, K. Impact of the Electromagnetic Brake Position on the Flow Structure in a Slab Continuous Casting Mold: An Experimental Parameter Study. *Metall. Mater. Trans. B* **2019**, *51*, 61–78. [CrossRef]
4. Thomas, B.; Singh, R.; Vanka, S.; Timmel, K.; Eckert, S.; Gerbeth, G. Effect of Single-Ruler Electromagnetic Braking (EMBr) Location on Transient Flow in Continuous Casting. *J. Manuf. Sci. Prod.* **2015**, *15*, 93–104. [CrossRef]
5. Timmel, K.; Eckert, S.; Gerbeth, G.; Stefani, F.; Wondrak, T. Experimental Modeling of the Continuous Casting Process of Steel Using Low Melting Point Metal Alloys—The LIMMCAST Program. *ISIJ Int.* **2010**, *50*, 1134–1141. [CrossRef]
6. Timmel, K.; Shevchenko, N.; Röder, M.; Anderhuber, M.; Gardin, P.; Eckert, S.; Gerbeth, G. Visualization of Liquid Metal Two-phase Flows in a Physical Model of the Continuous Casting Process of Steel. *Metall. Mater. Trans. B* **2015**, *46*, 700–710. [CrossRef]
7. Wondrak, T.; Eckert, S.; Gerbeth, G.; Klotsche, K.; Stefani, F.; Timmel, K.; Peyton, A.; Terzija, N.; Yin,W. Combined Electromagnetic Tomography for Determining Two-phase Flow Characteristics in the Submerged Entry Nozzle and in the Mold of a Continuous Casting Model. *Metall. Mater. Trans. B* **2011**, *42*, 1201–1210. [CrossRef]
8. Birjukovs, M.; Dzelme, D.; Jakovics, A.; Thomsen, K.; Trtik, P. Phase boundary dynamics of bubble flow in a thick liquid metal layer under an applied magnetic field. *Phys. Rev. Fluids* **2020**, *5*. [CrossRef]
9. Zhang, C. Liquid Metal Flows Driven by Gas Bubbles in a Static Magnetic Field. Ph.D. Thesis, Technischen Universität Dresden, Dresden, Germany, 2009.
10. Strumpf, E. Experimental study on rise velocities of single bubbles in liquid metal under the influence of strong horizontal magnetic fields in a flat vessel. *Int. J. Multiph. Flow* **2017**, *97*, 168–185. [CrossRef]
11. Zhang, C.; Eckert, S.; Gerbeth, G. Experimental Experimental study of single bubble motion in a liquid metal column exposed to a DC magnetic field. *Int. J. Multiph. Flow* **2005**, *31*, 824–842. [CrossRef]
12. Wang, Z.; Wang, S.; Meng, X.; Ni, M. UDV measurements of single bubble rising in a liquid metal Galinstan with a transverse magnetic field. *Int. J. Multiph. Flow* **2017**, *94*, 201–208. [CrossRef]
13. Shew, W.L.; Poncet, S.; Pinton, J.F. Force measurements on rising bubbles. *J. Fluid Mech.* **2006**, *569*, 51–60. [CrossRef]
14. Richter, T.; Keplinger, O.; Shevchenko, N.; Wondrak, T.; Eckert, K.; Eckert, S.; Odenbach, S. Single bubble rise in GaInSn in a horizontal magnetic field. *Int. J. Multiph. Flow* **2018**, *104*, 32–41. [CrossRef]
15. Schwarz, S. An Immersed Boundary Method for Particles and Bubbles in Magnetohydrodynamic Flows. Ph.D. Thesis, Technischen Universität Dresden, Dresden, Germany, 2014.
16. Schwarz, S.; Fröhlich, J. Numerical study of single bubble motion in liquid metal exposed to a longitudinal magnetic field. *Int. J. Multiph. Flow* **2014**, *62*, 134–151. [CrossRef]
17. Jin, K.; Kumar, P.; Vanka, S.; Thomas, B. Rise of an argon bubble in liquid steel in the presence of a transverse magnetic field. *Phys. Fluids* **2016**, *28*, 093301. [CrossRef]
18. Zhang, J.; Ni, M.J. Direct simulation of single bubble motion under vertical magnetic field: Paths and wakes. *Phys. Fluids* **2014**, *26*, 102102. [CrossRef]
19. Zhang, J.; Ni, M.J.; Moreau, R. Rising motion of a single bubble through a liquid metal in the presence of a horizontal magnetic field. *Phys. Fluids* **2016**, *28*, 032101. [CrossRef]

20. Wang, X.; Klaasen, B.; Degrève, J.; Mahulkar, A.; Heynderickx, G.; Reyniers, M.F.; Blanpain, B.; Verhaeghe, F. Volume-of-fluid simulations of bubble dynamics in a vertical Hele-Shaw cell. *Phys. Fluids* **2016**, *28*, 053304. [CrossRef]

21. Roig, V.; Roudet, M.; Risso, F.; Billet, A.M. Dynamics of a high-Reynolds-number bubble rising within a thin gap. *J. Fluid Mech.* **2012**, *707*, 444–466. [CrossRef]

22. Gaudlitz, D.; Adams, N. Numerical investigation of rising bubble wake and shape variations. *Phys. Fluids* **2009**, *21*, 122102. [CrossRef]

23. Mougin, G.; Magnaudet, J. Path Instability of a Rising Bubble. *Phys. Rev. Lett.* **2002**, *88*, 014502. [CrossRef]

24. Tripathi, M.; Sahu, K.; Govindarajan, R. Dynamics of an initially spherical bubble rising in quiescent liquid. *Nat. Commun.* **2015**, *6*, 6268. [CrossRef]

25. Zhang, J.; Ni, M.J. What happens to the vortex structures when the rising bubble transits from zigzag to spiral? *J. Fluid Mech.* **2017**, *828*, 353–373. [CrossRef]

26. Zhang, J.; Sahu, K.; Ni, M.J. Transition of bubble motion from spiralling to zigzagging: A wake-controlled mechanism with a transverse magnetic field. *Int. J. Multiph. Flow* **2020**, *136*, 103551. [CrossRef]

27. Will, J.; Mathai, V.; Huisman, S.; Lohse, D.; Sun, C.; Krug, D. Kinematics and dynamics of freely rising spheroids at high Reynolds numbers. *J. Fluid Mech.* **2021**, *912*, A16. [CrossRef]

28. Keplinger, O.; Shevchenko, N.; Eckert, S. Experimental investigation of bubble breakup in bubble chains rising in a liquid metal. *Int. J. Multiph. Flow* **2019**, *116*, 39–50. [CrossRef]

29. Keplinger, O.; Shevchenko, N.; Eckert, S. Visualization of bubble coalescence in bubble chains rising in a liquid metal. *Int. J. Multiph. Flow* **2018**, *105*, 159–169. [CrossRef]

30. Ziegenhein, T.; Lucas, D. Observations on bubble shapes in bubble columns under different flow conditions. *Exp. Therm. Fluid Sci.* **2017**, *85*, 248–256. [CrossRef]

31. Haas, T.; Schubert, C.; Eickhoff, M.; Pfeifer, H. A Review of Bubble Dynamics in Liquid Metals. *Metals* **2021**, *11*, 664. [CrossRef]

32. Liu, Z.; Li, B. Large-Eddy Simulation of Transient Horizontal Gas–Liquid Flow in Continuous Casting Using Dynamic Subgrid-Scale Model. *Metall. Mater. Trans. B* **2017**, *48*, 1833–1849. [CrossRef]

33. Yang, W.; Luo, Z.; Zhao, N.; Zou, Z. Numerical Analysis of Effect of Initial Bubble Size on Captured Bubble Distribution in Steel Continuous Casting Using Euler–Lagrange Approach Considering Bubble Coalescence and Breakup. *Metals* **2020**, *10*, 1160. [CrossRef]

34. Yang, W.; Luo, Z.; Gu, Y.; Liu, Z.; Zou, Z. Numerical Analysis of Effect of Operation Conditions on Bubble Distribution in Steel Continuous Casting Mold with Advanced Bubble Break-up and Coalescence Models. *ISIJ Int.* **2020**, *60*, 2234–2245. [CrossRef]

35. Taborda, M.; Sommerfeld, M.; Muniz, M. LES-Euler/Lagrange modelling of bubble columns considering mass transfer, chemical reactions and effects of bubble dynamics. *Chem. Eng. Sci.* **2021**, *229*, 116121. [CrossRef]

36. Akashi, M.; Keplinger, O.; Shevchenko, N.; Anders, S.; Reuter, M. X-ray Radioscopic Visualization of Bubbly Flows Injected Through a Top Submerged Lance into a Liquid Metal. *Metall. Mater. Trans. B* **2019**, *51*, 124–139. [CrossRef]

37. Saito, Y.; Mishima, K.; Tobita, Y.; Suzuki, T.; Matsubayashi, M. Measurements of liquid-metal two-phase flow by using neutron radiography and electrical conductivity probe. *Exp. Therm. Fluid Sci.* **2005**, *29*, 323–330. [CrossRef]

38. Saito, Y.; Mishima, K.; Tobita, Y.; Suzuki, T.; Matsubayashi, M.; Lim, I.C.; Cha, J.E. Application of high frame-rate neutron radiography to liquid-metal two-phase flow research. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **2005**, *542*, 168–174.

39. Lappan, T.; Sarma, M.; Heitkam, S.; Trtik, P.; Mannes, D.; Eckert, K.; Eckert, S. Neutron radiography of particle-laden liquid metal flow driven by an electromagnetic induction pump. *Magnetohydrodynamics* **2020**, *56*, 167–176. [CrossRef]

40. Sarma, M.; Ščepanskis, M.; Jakovics, A.; Thomsen, K.; Nikoluškins, R.; Vontobel, P.; Beinerts, T.; Bojarevics, A.; Platacis, E. Neutron Radiography Visualization of Solid Particles in Stirring Liquid Metal. *Phys. Procedia* **2015**, *69*, 457–463. [CrossRef]

41. Ščepanskis, M.; Sarma, M.; Vontobel, P.; Trtik, P.; Thomsen, K.; Jakovics, A.; Beinerts, T. Assessment of Electromagnetic Stirrer Agitated Liquid Metal Flows by Dynamic Neutron Radiography. *Metall. Mater. Trans. B* **2017**, *48*, 1045–1054. [CrossRef]

42. Dzelme, V.; Jakovics, A.; Vencels, J.; Köppen, D.; Baake, E. Numerical and experimental study of liquid metal stirring by rotating permanent magnets. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Hyogo, Japan, 14–18 October 2018; Volume 424, p. 012047. [CrossRef]

43. Birjukovs, M.; Dzelme, V.; Jakovics, A.; Thomsen, K.; Trtik, P. Argon bubble flow in liquid gallium in external magnetic field. *Int. J. Appl. Electromagn. Mech.* **2020**, *63*, S51–S57. [CrossRef]

44. Liu, L.; Keplinger, O.; Ziegenhein, T.; Shevchenko, N.; Eckert, S.; Yan, H.; Lucas, D. Euler–Euler modeling and X-ray measurement of oscillating bubble chain in liquid metals. *Int. J. Multiph. Flow* **2018**, *110*, 218–237. [CrossRef]

45. Krull, B.; Strumpf, E.; Keplinger, O.; Shevchenko, N.; Fröhlich, J.; Eckert, S.; Gerbeth, G. Combined experimental and numerical analysis of a bubbly liquid metal flow. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Dresden, Germany, 19–20 September 2017; Volume 228, p. 012006. [CrossRef]

46. Keplinger, O.; Shevchenko, N.; Eckert, S. Validation of X-ray radiography for characterization of gas bubbles in liquid metals. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Dresden, Germany, 19–20 September 2017; Volume 228, p. 012009. [CrossRef]

47. Liu, Y.; Ersson, M.; Liu, H.; Jönsson, P.; Gan, Y. A Review of Physical and Numerical Approaches for the Study of Gas Stirring in Ladle Metallurgy. *Metall. Mater. Trans. B* **2018**, *50*, 555–577. [CrossRef]

48. Cao, Q.; Nastac, L. Numerical modelling of the transport and removal of inclusions in an industrial gas-stirred ladle. *Ironmak. Steelmak.* **2018**, *45*, 984–991. [CrossRef]

49. Ramasetti, E.; Visuri, V.V.; Sulasalmi, P.; Palovaara, T.; Gupta, A.; Fabritius, T. Physical and CFD Modeling of the Effect of Top Layer Properties on the Formation of Open-Eye in Gas-Stirred Ladles With Single and Dual-Plugs. *Steel Res. Int.* **2019**, *90*, 1900088. [CrossRef]

50. Lou, W.; Zhu, M. Numerical Simulation of Desulfurization Behavior in Gas-Stirred Systems Based on Computation Fluid Dynamics–Simultaneous Reaction Model (CFD–SRM) Coupled Model. *Metall. Mater. Trans. B* **2014**, *45*, 1706–1722. [CrossRef]

51. Kusuno, H.; Sanada, T. Wake-induced lateral migration of approaching bubbles. *Int. J. Multiph. Flow* **2021**, *139*, 103639. [CrossRef]

52. Zhang, J.; Chen, L.; Ni, M.J. Vortex interactions between a pair of bubbles rising side by side in ordinary viscous liquids. *Phys. Rev. Fluids* **2019**, *4*, 043604. [CrossRef]

53. Zhang, J.; Ni, M.J.; Magnaudet, J. Three-Dimensional Dynamics of a Pair of Deformable Bubbles Rising Initially in Line. Part 1: Moderately Inertial Regimes. *J. Fluid Mech* **2021**, *920*, A16. [CrossRef]

54. Filella, A.; Ern, P.; Véronique, R. Interaction of two oscillating bubbles rising in a thin-gap cell: Vertical entrainment and interaction with vortices. *J. Fluid Mech.* **2020**, *888*, A13. [CrossRef]

55. Klevs, M.; Birjukovs, M.; Zvejnieks, P.; Jakovics, A. Dynamic mode decomposition of magnetohydrodynamic bubble chain flow in a rectangular vessel. *Phys. Fluids* **2021**, *33*, 083316. [CrossRef]

56. Lehmann, E.; Vontobel, P.; Wiezel, L. Properties of the radiography facility NEUTRA at SINQ and its use as European reference facility. *Nondestruct. Test. Eval.* **2001**, *16*, 191–202. [CrossRef]

57. Kaestner, A.; Hartmann, S.; Kühne, G.; Frei, G.; Grünzweig, C.; Josic, L.; Schmid, F.; Lehmann, E. The ICON beamline—A facility for cold neutron imaging at SINQ. *Nucl. Instrum. Methods Phys. Res. Sect. A-Accel. Spectrometers Detect. Assoc. Equip.* **2011**, *659*, 387–393. [CrossRef]

58. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear total variation based noise removal algorithms. *Phys. D Nonlinear Phenom.* **1992**, *60*, 259–268. [CrossRef]

59. Clift, R.; Grace, J.; Weber, M. *Bubbles, Drops, and Particles*; Academic Press: New York, NY, USA, 1978.

60. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 629–639. [CrossRef]

61. Weickert, J.; Romeny, B.; Viergever, M. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Process.* **1998**, *7*, 398–410. [CrossRef] [PubMed]

62. Wolfram Research. PeronaMalikFilter. Available online: https://reference.wolfram.com/language/ref/PeronaMalikFilter.html (accessed on 16 October 2021).

63. Le, T.; Chartrand, R.; Asaki, T. A Variational Approach to Reconstructing Images Corrupted by Poisson Noise. *J. Math. Imaging Vis.—JMIV* **2007**, *27*, 257–263. [CrossRef]

64. Wolfram Research. TotalVariationFilter. Available online: https://reference.wolfram.com/language/ref/TotalVariationFilter.html (accessed on 16 October 2021).

65. Sapiro, G. Vector (Self) Snakes: A Geometric Framework for Color, Texture, and Multiscale Image Segmentation. In Proceedings of 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 1 September 1996; Volume 1, pp. 817–820. [CrossRef]

66. Wolfram Research. CurvatureFlowFilter. Available online: https://reference.wolfram.com/language/ref/CurvatureFlowFilter.html (accessed on 16 October 2021).

67. Tone Reproduction. *The Reproduction of Colour*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2004; Chapter 6, pp. 47–67. [CrossRef]

68. Gonzalez, R.; Woods, R. *Digital Image Processing*; Prentice-Hall, Inc.: Saddle River, NJ, USA, 2006.

69. Haralick, R.; Sternberg, S.; Zhuang, X. Image Analysis Using Mathematical Morphology. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9*, 532–550. [CrossRef]

70. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]

71. Canny, J. A Computational Approach To Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]

72. Getreuer, P. Chan–Vese Segmentation. *Image Process. Line* **2012**, *2*, 214–224. [CrossRef]

73. Antonov, A. Quantile Regression with B-Splines. 2014. Available online: https://mathematicaforprediction.wordpress.com/2014/01/01/quantile-regression-with-b-splines/ (accessed on 16 October 2021).

74. Wolfram Research. Inpaint. Available online: https://reference.wolfram.com/language/ref/Inpaint.html (accessed on 16 October 2021).

75. Zvejnieks, P.; Birjukovs, M.; Klevs, M.; Akashi, M.; Eckert, S.; Jakovics, A. MHT-X: Offline Multiple Hypothesis Tracking with Algorithm X. *arXiv* **2020**, arXiv:2101.05202.