UCRL-TR-208853

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Feature Extraction from Simulations and Experiments: Preliminary Results Using a Fluid Mix Problem

C. Kamath, T. Nguyen

January 5, 2005

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Feature Extraction from Simulations and Experiments: Preliminary Results Using a Fluid Mix Problem

Chandrika Kamath
Thinh Nguyen
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
kamath2@llnl.gov

February 2, 2005

1 Introduction

Code validation, or comparing the output of computer simulations to experiments, is necessary to determine which simulation is a better approximation to an experiment. It can also be used to determine how the input parameters in a simulation can be modified to yield output that is closer to the experiment.

In this report, we discuss our experiences in the use of image processing techniques for extracting features from 2-D simulations and experiments. These features can be used in comparing the output of simulations to experiments, or to other simulations. We first describe the problem domain and the data. We next explain the need for cleaning or denoising the experimental data and discuss the performance of different techniques. Finally, we discuss the features of interest and describe how they can be extracted from the data.

The focus in this report is on extracting features from experimental and simulation data for the purpose of code validation; the actual interpretation of these features and their use in code validation is left to the domain experts.

2 Problem Description

We conducted our study in the use of image processing for code validation using a fluid mix problem. We consider the case of Richtmyer-Meshkov instability which results when an impulsive acceleration is applied to the interface separating two fluids of different densities, for example, as a result of a shock wave striking the interface perpendicularly. Such instabilities arise in diverse situations such as supernovas, oceans, and supersonic combustion, and are therefore the subject of much research.

The experimental data used in this study were obtained from Jeff Jacobs at the Experimental Fluid Dynamics and Instability Laboratory, Department of Aerospace and Mechanical Engineering, at the University of Arizona. They were generated using a shock-tube containing a column of acetone/air mixture on the top and a column of sulphur hexachloride mixture at the bottom, as described in [1]. The data discussed in this report are the re-shock experiments obtained when the original shock wave at Mach 1.3 is reflected off the lower wall of the shock tube. The images from the experiment are given in Fig. (1), which shows the mixing of the two fluids at various time steps.

The simulation data used in this study were obtained from the Raptor code, which is a multi-material Eulerian adaptive mesh refinement code by Jeff Greenough, from AX Division at LLNL. Fig. (2) shows the images from the simulation as it evolves over time. Some of the images have been cropped at the top and bottom as they are larger than other images (see Table 1). The filenames of the experimental and simulation data, as well as the sizes of the images is given in Table 1.

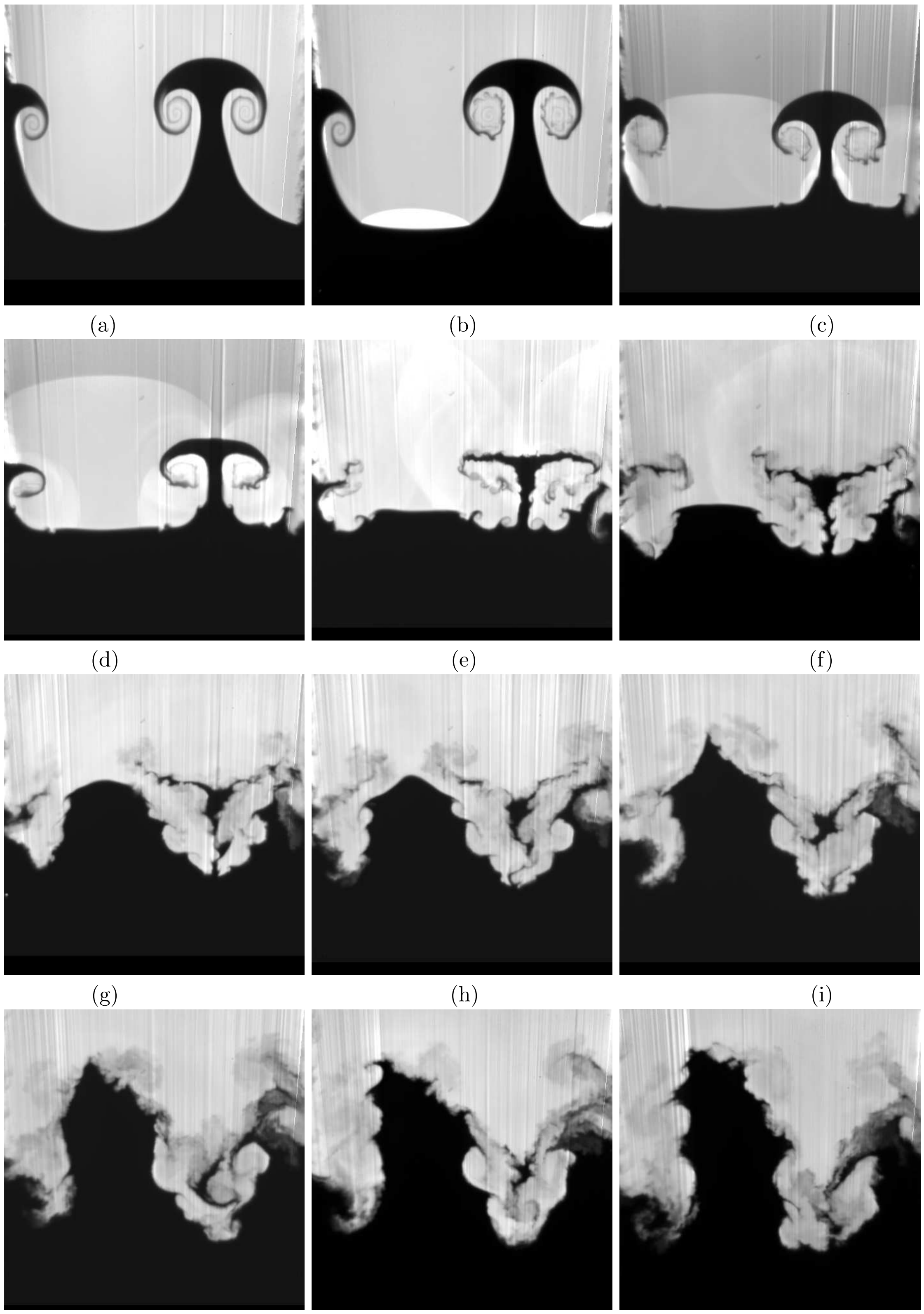

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 1: The experimental images.

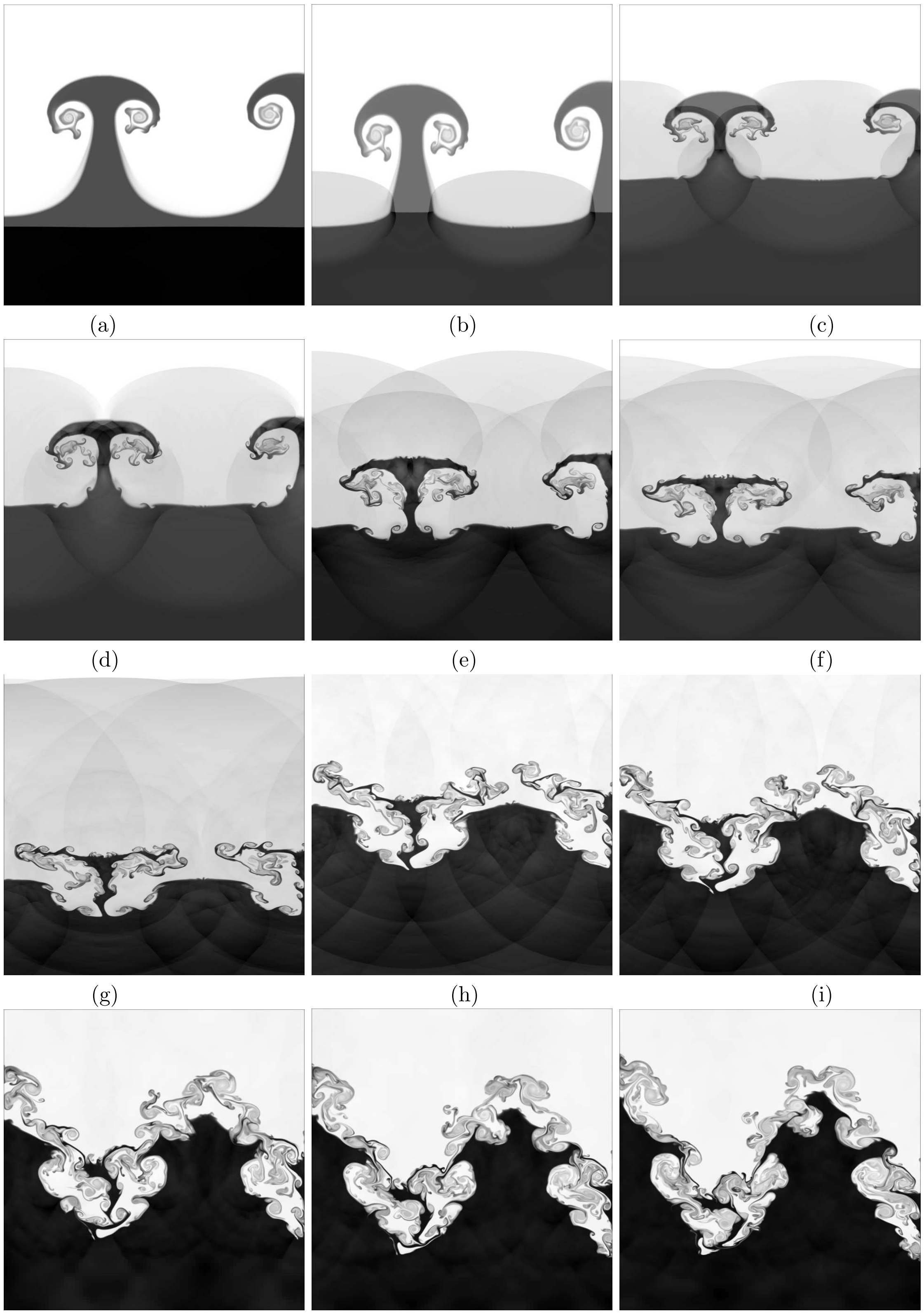


Figure 2: The simulation images.

File name (experiment)	image size	File name (simulation)	image size
T1_12	469 × 469	1325	1280 × 1536
T2_27	461 × 461	1335	1280 × 1280
T3_17	464 × 463	1345	1280 × 1536
T4_18	462 × 462	1350	1280 × 1344
T5_19	467 × 467	1360	1280 × 1728
T6_20	467 × 467	1365	1280 × 1664
T7_21	465 × 465	1380	1280 × 2176
T8_22	467 × 467	1395	1280 × 1984
T9_23	468 × 468	1405	1280 × 2048
T10_24	468 × 468	1420	1280 × 1408
T11_25	468 × 468	1435	1280 × 1472
T12_26	469 × 469	1450	1280 × 1536

Table 1: Image sizes for the experimental and simulation data. Note the variation in the size of the simulation data.

The goal of this study is to extract features from the experimental and simulation data that will enable us to compare the two. To accomplish this, we chose two different approaches, one for the early time steps, where the mushroom-shaped "objects" can be clearly identified in the images, and the second approach for the later time steps, where the fluid mixing is at an advanced stage and no clear objects can be identified in the images.

For the early time steps, we can characterize the mushroom using features such as the height of the mushroom, the width of the cap, the height of the cap, and the width of the stem, as shown in Fig. (3). However, we note that the experimental images are noisy, with structured noise in the form of vertical lines. If these vertical "noise edges" are weaker than the edges of interest in the mushroom, it is possible to use a simple edge detector with a suitable threshold to separate the mushroom from the background, Fig. (4a). However, when the noise edges are of the same strength as the mushroom edges, the edge detector also picks up the lines, making feature extraction from the edge image difficult, Fig. (4b). Hence, our first step was to denoise the experimental images.

3 Denoising experimental images

General image denoising techniques have been well explored for many years. Most of these techniques, however, are designed to remove random noise such as Gaussian noise or salt and pepper noise and are therefore less effective when the noise does not appear randomly in either pixel value or location. The noise in the experimental images, being structured, does not lend itself to the more traditional denoising techniques. Consider for example, one of the experimental images, Figure (5a). To remove the vertical noise lines, we first applied two horizontal median filters of two different sizes to this image. Figure (5b) shows the resulting image using median filter of size 1x5. Note that the lines are still visible. Using a filter size of 1x10, the lines are removed; however, other details of the image are also removed as shown in Figure (5c). For this image, the small-size median filter is ineffective for removing the lines since the width of these lines is too large for a small-size filter to find the right median pixel value. On the other hand, when a large-size median filter is used, the resulting image is blurred significantly. Similarly, using a Wiener filter with large enough size, e.g. 1x10, results in a blurred image as shown in Figure (5d).

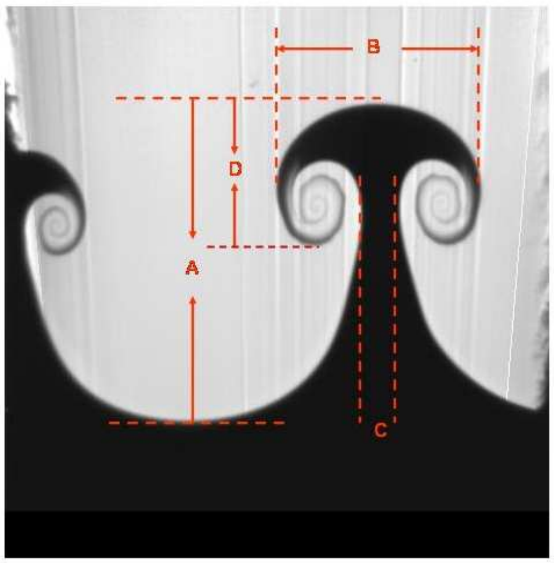


Figure 3: The features used to represent the mushroom include (A): the height of the mushroom, (B): the width of the cap, (C): the width of the stem, and (D): the height of the cap.

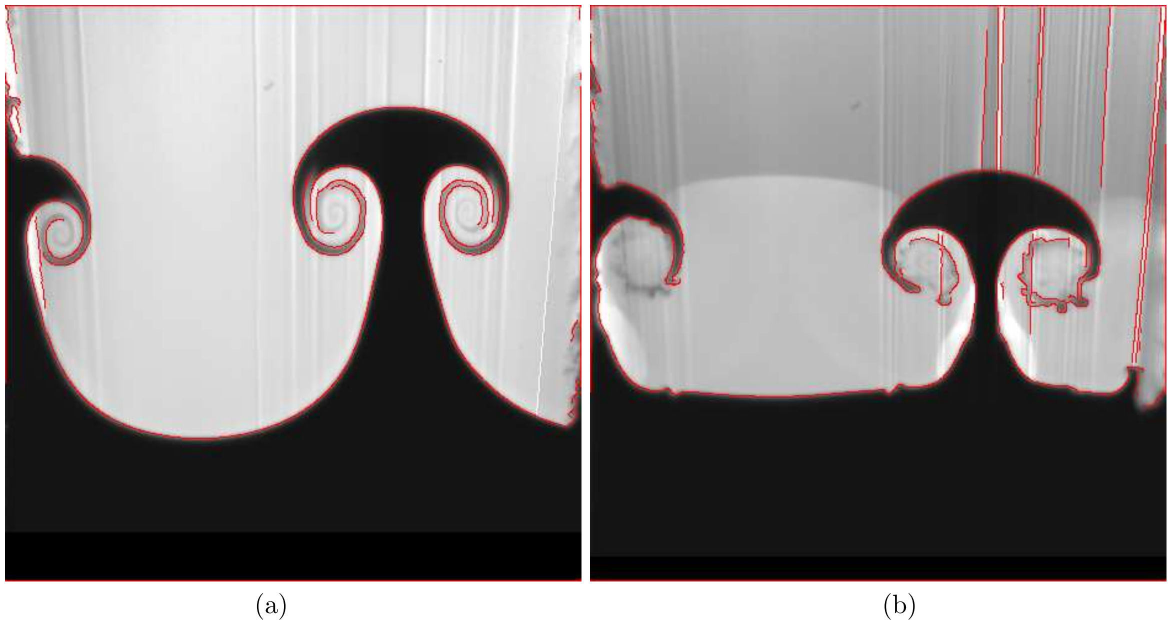
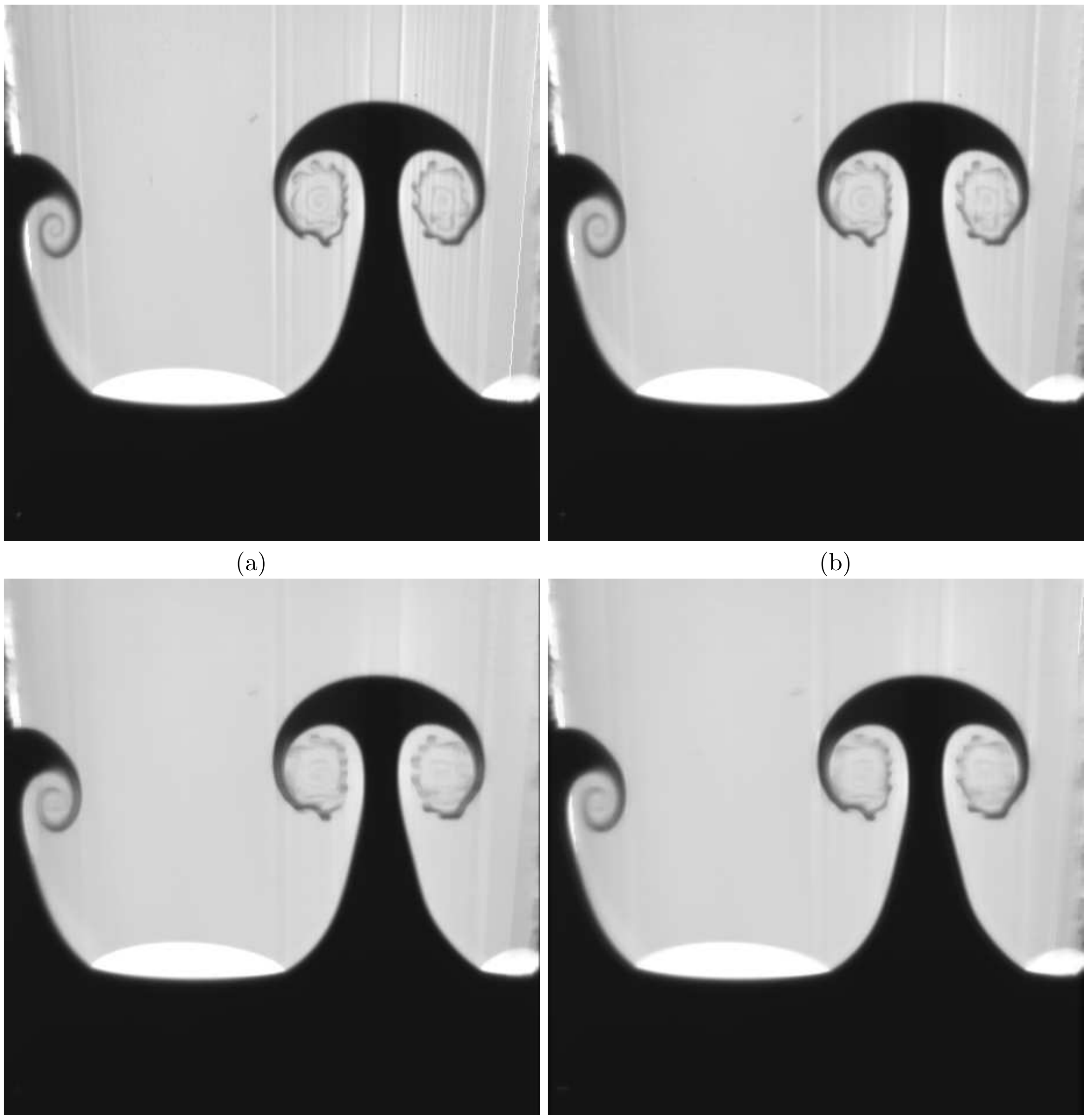


Figure 4: The need for denoising experimental images: The vertical noise lines in the experimental images can be removed by suitable thresholding in the edge detector if they are weak (Panel (a)). But, if the edges of the noise lines are the same strength as the edges of the mushroom, a simple setting of the threshold in the edge detector cannot remove these lines (Panel (b)). Edges in the images are obtained using the Canny edge detector, and superposed in red on the original image.



(a) (b) (c) (d)

Figure 5: Application of traditional denoising techniques. (a) The original experimental image with structured noise as black and white lines. Images resulting from the application of (b) median filter of size one pixel down and five pixel across (1x5); (c) median filter of size one pixel down and ten pixels across (1x10); (d) Wiener filter of size 1x10.

3.1 Technical Approach for Denoising

We next describe an approach which removes structured noise in the form of lines in an image by exploiting the characteristics of a line to reduce the blurring side effect. As discussed previously, traditional denoising techniques fail to remove the lines in the original image. These lines have substantial width, and are structured enough to be considered as features rather than noise by most of the denoising algorithms. To remove these lines properly, an algorithm should be able to differentiate between the unwanted lines and the rest of the image.

To accomplish this, we use a two-step approach: In the first step, we segment the image to identify the regions occupied by the unwanted lines. We use an edge detector to find the edges in the image, followed by a bounding ratio test (which will be discussed shortly) to differentiate between the unwanted lines and other features in the image. In the second step, we apply appropriate techniques to remove these lines locally. We focus only on the segmented regions found in the first step and thus avoid the adverse affect of smoothing the entire image.

To implement this approach, we first need to select a segmentation algorithm that can detect lines of different widths and intensities. In our proposed approach, we use the Canny edge detector for extracting the edges and additional heuristic methods for capturing the width of the unwanted lines. Next, we need to select the line removal algorithm, which determines how to fill the line regions with the appropriate pixel values such that the resulting image appears as natural as possible. In this case, a median filter with an appropriate width proved to be an effective method.

3.2 Algorithm

We next present the details of the algorithm, along with an explanation for each step. We show how the original image with the unwanted vertical lines representing the noise, Figure (6a), is modified after the application of each step in our approach.

A: Segmentation steps

- *Step 1* - Enhance the image using the following sharpening filter:

$$\begin{vmatrix} -0.6667 & -1.6667 & -0.6667 \\ -1.6667 & +4.3333 & -0.6667 \\ -0.6667 & -1.6667 & -0.6667 \end{vmatrix}$$

  to produce the *sharp* image. The sharpening filter enhances the image so that the edge detector can differentiate the edges more effectively. Further, it can enhance the contrast, resulting in an image that is visually better as shown in Figure (6b).

- *Step 2* - Apply the Canny edge detector with the lower and upper thresholds of 0.001 and 0.015, respectively, to the *sharp* image to produce a binary mask. These are the two thresholds used by the Matlab implementation of the Canny edge detector to implement hysteresis thresholding. This step locates all the edge pixels in the image, as shown in Figure (6c).

- *Step 3* - Segment the binary image into multiple objects. We use 8-connectivity of the edge pixels, that is, two pixels are considered to belong to the same object if they are adjacent to each other either horizontally, vertically, or diagonally. These objects include both the unwanted lines and other features in the image as shown in Figure (6d).

- *Step 4* - Differentiate between unwanted line objects and other objects in the image. This is done by considering the dimensions of the bounding box of each object. An object whose bounding box ratio of width/height is less than 0.1 is considered as a line. Since all the lines in the experimental images are almost vertical, we expect the bounding box ratio to be small for the lines, but not the other objects in the edge image. The threshold value of 0.1 is chosen heuristically based on the experimental images. Next, a binary *object* mask is created with its pixel values set to 1 if the pixel belongs to any line object and zero otherwise. This results in the image in Figure (6e).

- *Step 5* - Fill the region between the thin lines that are close to each other to create lines with width greater than 1. Thus, if two lines are spatially close to each other, e.g. 4 pixels apart, then the algorithm fills in all the pixels between the two thin lines with value 1. After this step, the binary *object* mask should appear to have thick lines represented by pixel value equal to 1. This merging step is done to account for the fact that the noise lines have a width greater than a single pixel. Applying the Canny edge detector to a thick line in the original image results in two thin lines representing the two edges of the original thick line. Since all the pixels between these thin lines are considered part of the unwanted thick line, we want to merge these two thin lines back into one thick line. Figure (6f) shows the resulting image after line merging.

The goal of steps 1 through 5 is to create an *object* mask containing the unwanted lines to be removed. In steps 6 to 9, we describe how we can remove these lines using the *object* mask so that the resulting image appears as natural as possible.

B: Line removal steps

- *Step 6* - Create a *smoothing* mask for subsequent boundary smoothing. In the *smoothing* mask, the values of pixels around the borders of the lines in the *object* mask are set to 2, the values of pixels belong to the lines are set to 1, and the values of other pixels are set to zero. This *smoothing* mask will be used in step 9.
- *Step 7* - Apply a median filter of size 5x15 to the original image to remove all the lines and to produce the *median* image. As mentioned earlier, this smooths the lines, but also blurs the rest of the image. Figure (6g) shows the resulting image.
- *Step 8* - Use the *smoothing* mask to remove the lines in the *sharp* image resulting in the *output* image. The rules for setting pixel values in the *output* image are as follows: If the value in the *smoothing* mask is 1, set the *output* image pixel to the value from the *median* image; if it is 0 or 2, set it to the value from the *sharp* image. This step ensures that only pixels within the unwanted lines are affected by the median filter. Hence the *output* image preserves the other features. Figure (6h). shows the image that results from this step. Since this step fills the pixel values of the *output* image from either the *median* or *sharp* images, discontinuities may arise at the borders of the lines. This problem is alleviated in step 9.
- *Step 9* - Apply an average filter of size 5x5 at those pixels whose value in the *smoothing* mask is 2. Recall that the values of the pixels around the line borders are set to 2 in step 7. This step smooths out the discontinuities that may arise at the borders of the lines after step 8. This results in the image shown in Figure (6i).

3.3 Improvements to the denoised image

In Figure (6i), the lines inside the spiral regions are still visible as the Canny edge detector cannot capture these lines. To remove these lines, we extend the lines in Figure (6e), through the spirals as seen in Figure (7a). This is done by calculating the slope of the line using the two extreme points on it and extending it downward. We then thicken the lines as in step 5 of the algorithm to produce an *extended line* mask containing the extended lines. Using this mask, we now apply a median filter of size 1x4 to the image pixels belonging to the mask, resulting in the *small median* image. The reason for using a small size median filter is to avoid destroying the small features in the spiral regions. The pixels in the new output image are then set to equal to the corresponding pixels in the *small median* image if they belong to the *extended line* mask only. Otherwise, they are set to the values of *output* image from the above algorithm. Figure (7b) shows the final image with the lines inside the spiral regions removed.

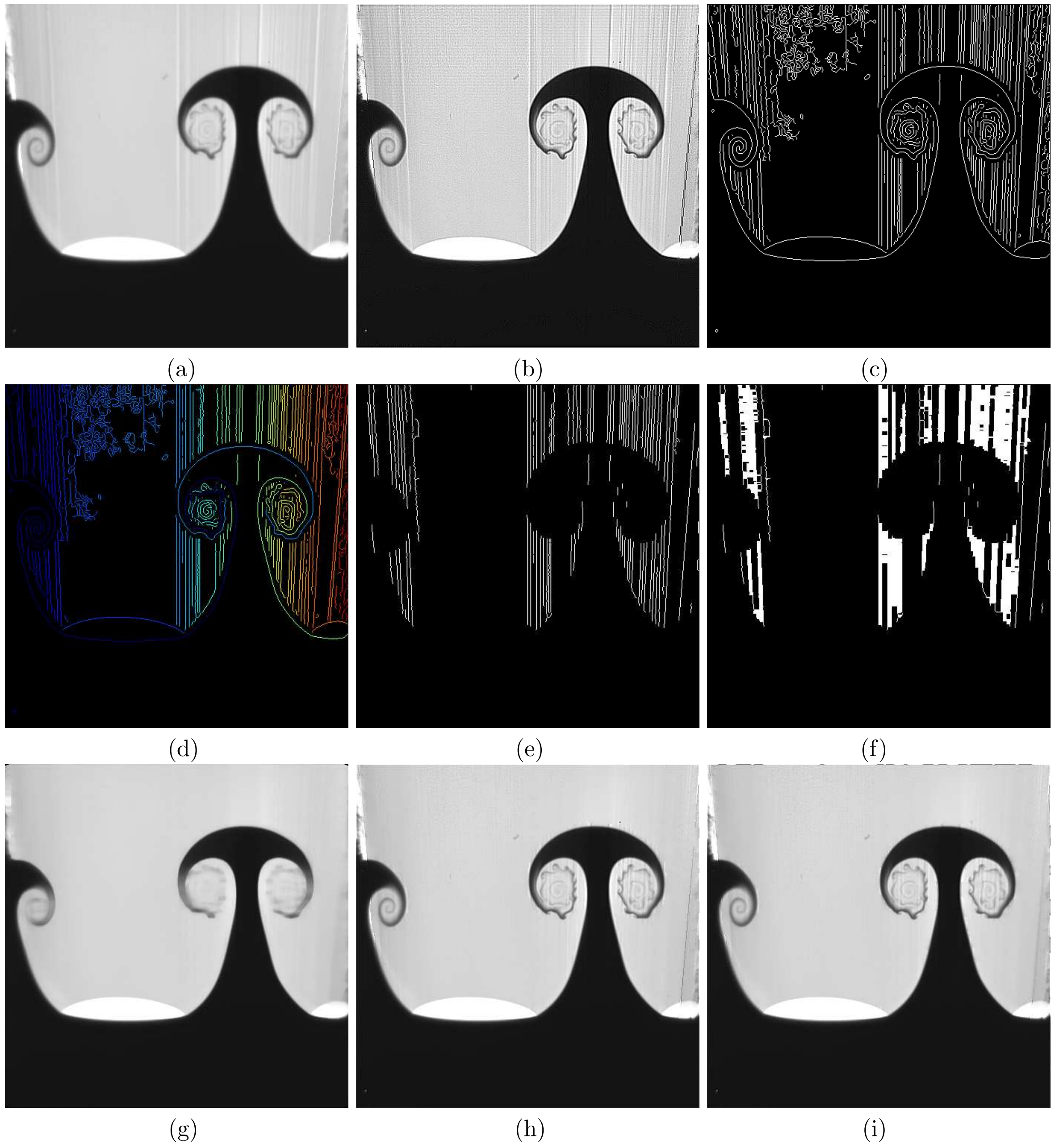


Figure 6: (a) Original image. (b) Sharp image. (c) Output of the Canny edge detector. (d) grouping the edges into objects as indicated by the different colors. (e) Mask of thin lines. (f) Mask of thick lines after merging. (g) Median image resulting from applying the median filter of 5x15. (h)Output image without border smoothing. (i) Output image with border smoothing.

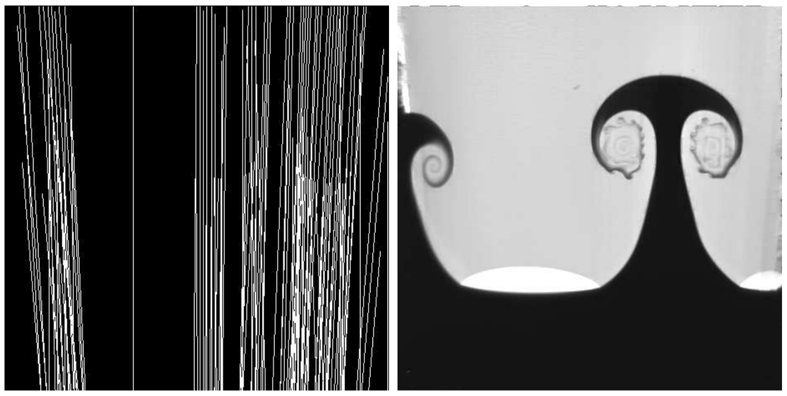

(a) (b)

Figure 7: Improvements to the denoising: (a) Mask with extended lines. (b) Final output image with the lines inside the spiral regions removed

4 Feature extraction from denoised experiments and simulations

In this section, we describe the approach used to extract features from the experimental and simulation data. As we mentioned before, the features extracted from the earlier time steps, when the objects in the data are still clearly identifiable, are different from those in the later time steps, when the fluid mixing is in a more advanced stage.

4.1 Feature extraction: early time steps

In Section 2, we described the features that can be used to characterize the mushroom-shaped objects in the images from the early time steps. Using an edge image, for example, in Figure (8), we can extract these features of interest by first identifying key points in the image as follows. We exploit the fact that if we count the number of edge pixels in each column of the image (hereafter abbreviated as NEPC), starting from the left edge and moving to the right, the changes in this number can be used to determine the points of interest.

- Point A: This can be used to obtain the height of the mushroom. It can be calculated as the point where the NEPC is 1 and the y-location of the edge pixel is minimum. Starting from the left edge, we first find the x-location where the NEPC is 1. Then, we keep track of the y-location of the edge pixel and identify the point where it reaches a minimum, before increasing again.
- Point B: This can be used to determine the width of the mushroom cap. When we move from point A to the right in the image, point B can be considered to be where the NEPC increases from 1 to a number greater than 1.
- Point C: This can be used to determine the width of the stem of the mushroom. Having identified point B, as we continue to move to the right in the image, point C can be considered to be the X location where the NEPC decreases from a number greater than 1 to 1.
- Point C': This, along with C, can be used to determine the width of the stem of the mushroom. Having identified point C, as we continue to move to the right in the image, point C' can be considered to be where the NEPC increases from 1 to a number greater than 1. Note that the y-location of this point may not necessarily be the same as the y-location of point C. Also, the width of the stem is defined as the shortest distance between lines tangential to the edge pixels in the stem.

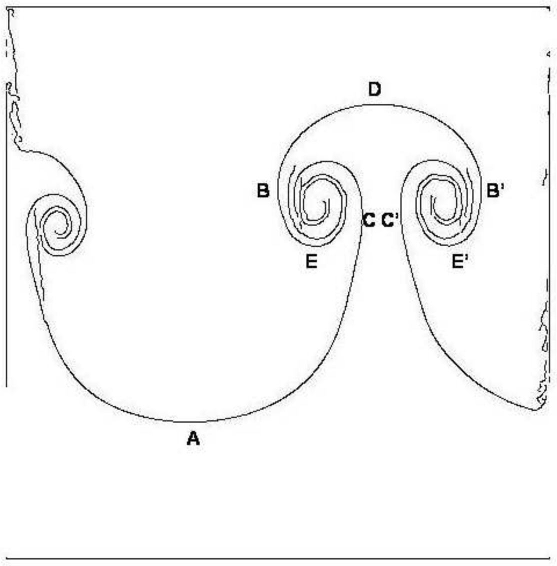


Figure 8: The points on the edge image that can be used to extract the features in Figure (3).

- Point B': This, along with B, can be used to determine the width of the mushroom cap. Having identified point C, as we continue to move to the right in the image, point C' can be considered to be where the NEPC decreases from a number greater than 1 to 1.
- Point D: this gives the height of the mushroom and is defined as the highest y-location of an edge pixel as we move to the right from B to B'.
- Points E and E': Point E (E') is considered to be the lowest point on the left (right) side of the mushroom cap. It can be calculated in a manner similar to the calculation of B, except that we now consider the number of edge pixels in a row segment of the image that stretches from the x-location of B (B') to the x-location of D. As this row-segment moves down, the point of interest is the y-location where the number of edge pixels reduces from a number greater than 1 to 1.

Figures (13) through (16) show four panels each with the original image, the cleaned image using the approach described in Section 3, the edge image, and the edge image superposed on the cleaned image. The edge image was obtained using the Canny edge detector and the same set of parameters was used on all four images.

The features extracted from the edge image for these four experimental images are given in Table 2, and the features for the mushroom are given in Table 3.

A similar approach to calculating the mushroom features can be used for the images from the simulation. However, since the simulation data captures the effects of the reshock better, especially in the region of the darker fluid, many more strong edges are obtained in the edge image. As a result, the calculation of the interest points has to be modified. For example, the calculation of B, B', C, and C' cannot be done by considering the points where the number of edge pixels in a column transitions to or from 1; instead we need to consider a larger number. This can be set either through experimentation or by including additional tests to confirm that the correct point has been identified. Also, the calculation of D as the highest point on the mushroom has to be done with care as the edges due to the reshock could be higher than the top of the mushroom.

Figures (17) and (18) show three panels each with the simulation image, the edge image, and the edge image superposed on the simulation. The edge image was obtained using the Canny edge detector and the same set of parameters was used on all four images. The features extracted from the edge

Data set	B_x	B'_x	C_x	C'_x	D_y	A_y	E_y	E'_y
Experiment 1	233	408	306	339	384	115	264	265
Experiment 2	231	412	309	338	376	115	252	253
Experiment 3	233	409	309	325	330	147	221	215
Experiment 4	238	409	313	333	309	168	232	229

Table 2: X or Y locations of the interest points on the mushroom corresponding to Figure (8) for the experimental images.

Data set	Height	Width of cap	Width of stem	Height of cap(L)	Height of cap (R)
Experiment 1	270	176	34	121	120
Experiment 2	262	182	30	124	124
Experiment 3	184	177	17	110	116
Experiment 4	142	172	21	78	81

Table 3: Mushroom features (in pixels) for the four experimental images.

Data set	B_x	B'_x	C_x	C'_x	D_y	A_y	E_y	E'_y
Simulation 1	182	664	381	470	1171	448	845	873
Simulation 2	179	667	381	470	938	328	609	633
Simulation 3	183	667	402	448	1090	636	829	841
Simulation 4	175	673	403	446	983	596	762	777

Table 4: X or Y locations of the interest points on the mushroom corresponding to Figure (8) for the simulation images.

Data set	Height	Width of cap	Width of stem	Height of cap (L)	Height of cap (R)
Simulation 1	724	483	90	327	299
Simulation 2	610	489	90	330	305
Simulation 3	455	485	47	262	250
Simulation 4	388	499	43	222	206

Table 5: Mushroom features (in pixels) for the four simulation images.

images for these four simulation images are given in Table 4, and the features for the mushroom are given in Table 5.

Note that a direct comparison of the values in Tables (3) and (5) is not possible as the pixels sizes are not the same between the experiments and the simulations. A possible option would be to consider ratios of quantities, or, if the pixels sizes are known for both the experimental and the simulation data, use them to calculate the mushroom features exactly.

4.2 Feature extraction: later time steps

At the later time steps, when the mushroom-shape is no longer clearly identifiable, we need to consider alternate features to represent each image. Figure 19 shows the cleaned versions of the experimental images at the later time steps. A feature that can be used to compare the later time steps is the probability distribution histogram of the image. However, given the large light and dark regions at the top and bottom of the images, using the PDF of the entire image may be misleading. A partial solution to this is to use the PDF of the mixing layer.

In our work, we consider two definitions of the mixing layer. First, for each y, we consider the average of values along the x direction. For low values of y (i.e. the bottom of the image), this average will be nearly 0 (i.e. all black), while for the very high values of y at the top of the image, this average will be nearly 255 (i.e. white, assuming 8-bit data). Panels (a) in Figures 20 through 35 show the plot of the x-average for the experimental and simulation images at the later time steps. We then use a 5% and 95% threshold on this plot to obtain the height of the mixing layer, as shown highlighted in panels (c) of the Figures. Panels (e) are the 64-bin PDFs of the mixing layer.

The second definition considers the variance along the x-direction, as shown in panels (b) of the figures. By thresholding on both the value of the variance (as a percentage of the difference between the maximum and minimum variance), and the gradient of the variance (calculated using central differencing), we obtain the mixing region as indicated in panels (d) of the figures. Panels (f) are the 64-bin PDFs of the mixing layer.

The same threshold values are used for both the experimental and the simulation data. The simulation data values are scale-quantized to lie between 0 and 255 before the calculation of the average or variance. Note that some of the experimental images have a wide black border (pixel intensity = 0) at the bottom, which appears to have been added to make the image square. This leads to a jump in the intensity as the darker fluid has intensity of around 20, resulting in an inaccurate delineation of the mixing layer when it is calculated using the average.

Once the mixing region has been identified for both the experiments and the simulations, the data values in the region are scale-quantized to lie between 0 and 255 before the PDF is obtained. Using a 64-bin PDF, the distributions of the mixing regions in the cleaned experimental images are given in Figures 9 and 10, where the mixing region is defined by the average and the variance, respectively. The corresponding distributions for the simulation data are given in Figures 11 and 12.

A direct comparison of the PDFs of the experimental images and the simulation data is valid only when the two images are of the same variable and have been processed identically prior to the extraction of the mixing region. This is not the case for the problem considered in this report. However, it is possible to use the approach outlined in this section to compare simulations to each other, using the same variable from both simulations.

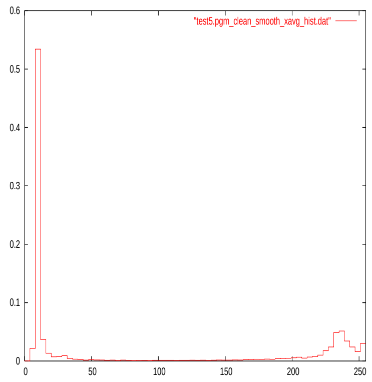
5 Acknowledgments

We gratefully acknowledge the domain expertise and technical insights provided by Paul Miller and Jeff Greenough of LLNL. We would like to thank Prof. Jeff Jacobs of the University of Arizona and Jeff Greenough of LLNL for access to the experimental and simulation data sets, respectively.

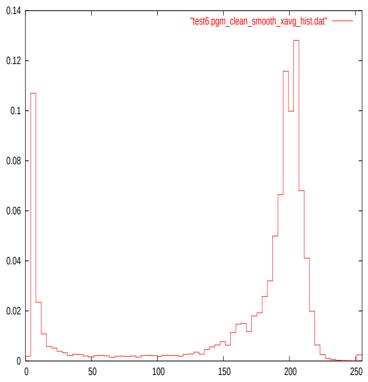
UCRL-TR-208853: This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

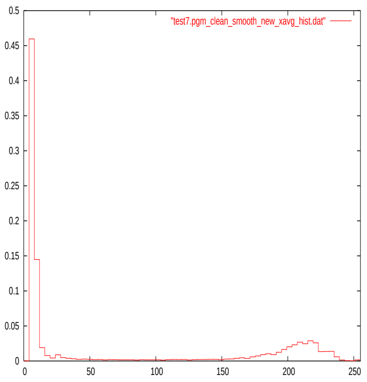
[1] Jacobs, J. W., and Collins, B. D. Experimental study of the Richtmyer-Meshkov instability of a diffuse interface. In *Proceedings of the 22-nd International Symposium on Shock Waves* (July 1999).



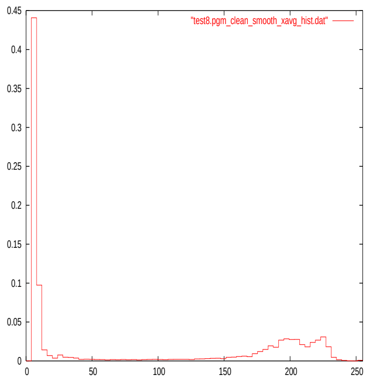
(a)



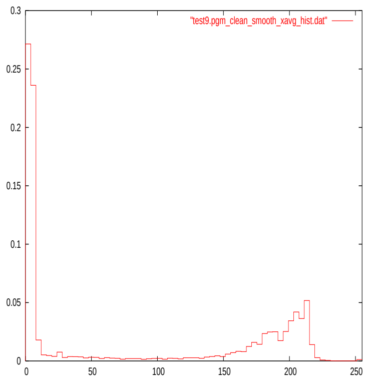
(b)



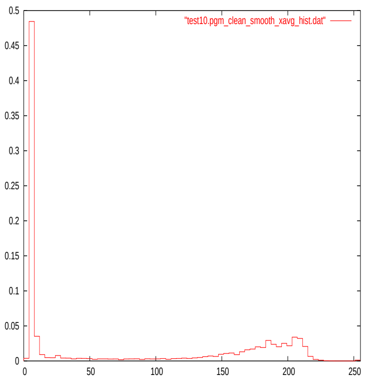
(c)



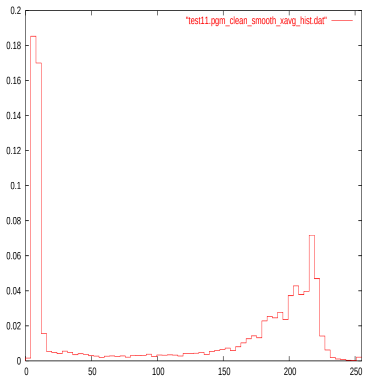
(d)



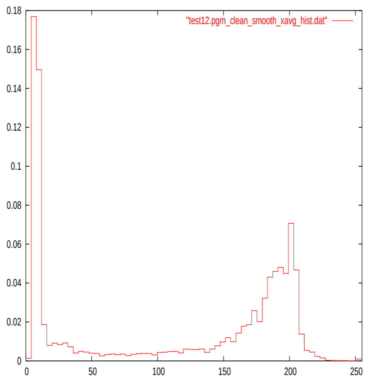
(e)



(f)



(g)



(h)

Figure 9: The probability distribution histograms of the mixing regions of the cleaned experimental images at the later time steps. These were obtained using the average along the x-direction for the mixing layer. The number of bins in the histogram is 64. Panels (a) through (h) are time steps 5 through 12.

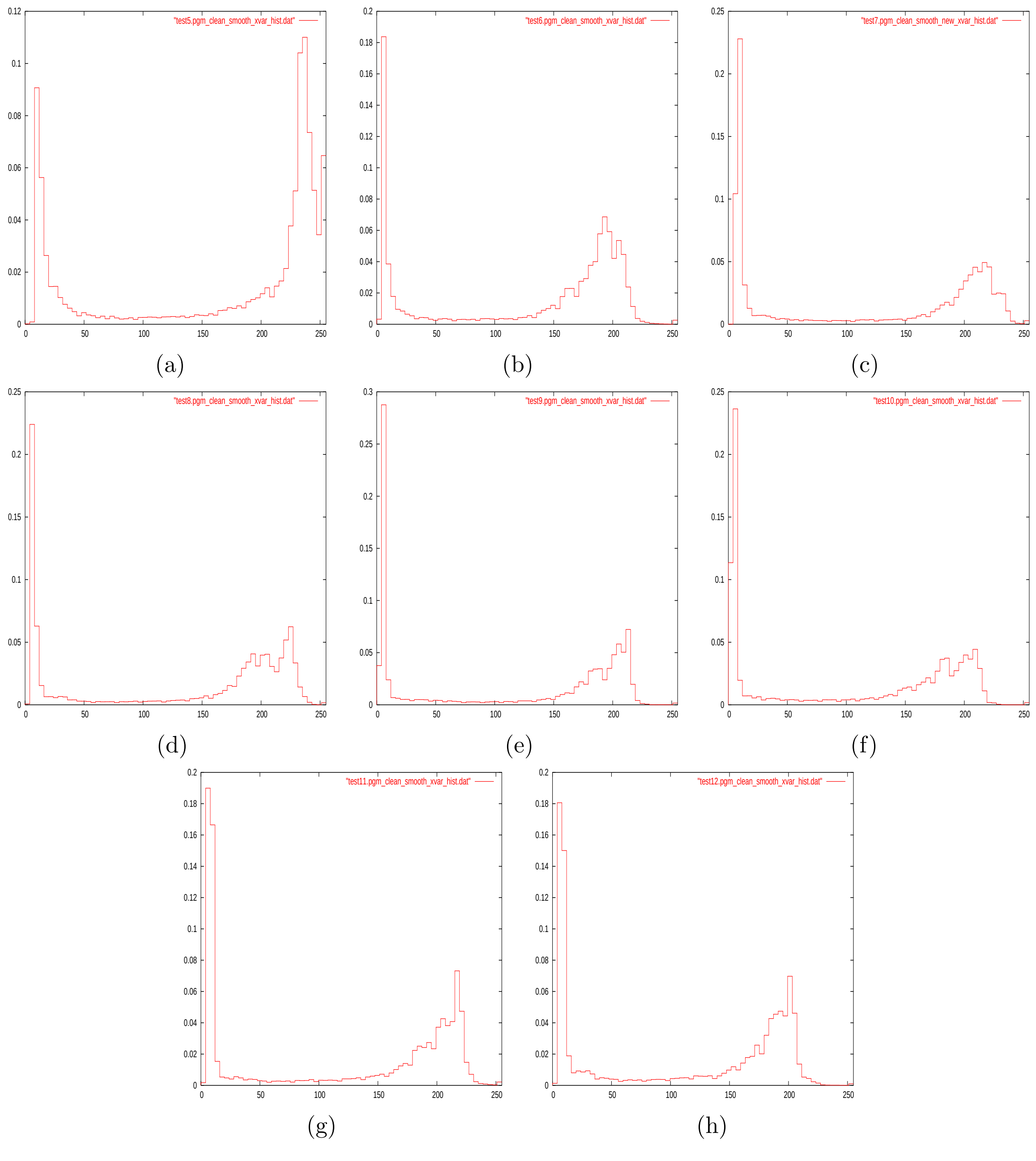
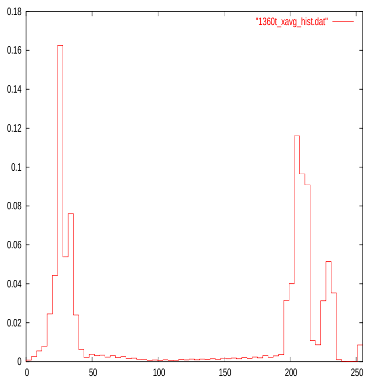
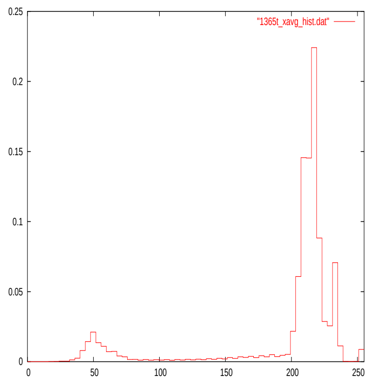


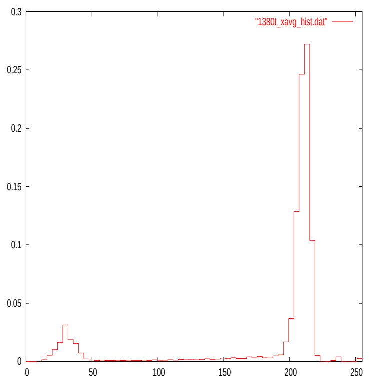
Figure 10: The probability distribution histograms of the mixing regions of the cleaned experimental images at the later time steps. These were obtained using the variance along the x-direction for the mixing layer. The number of bins in the histogram is 64. Panels (a) through (h) are time steps 5 through 12.



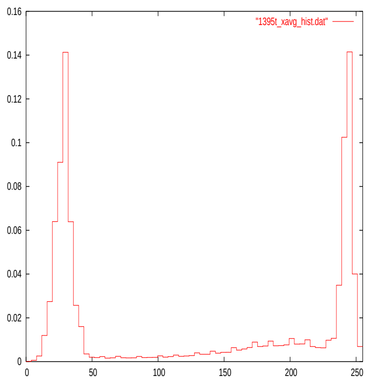
(a)



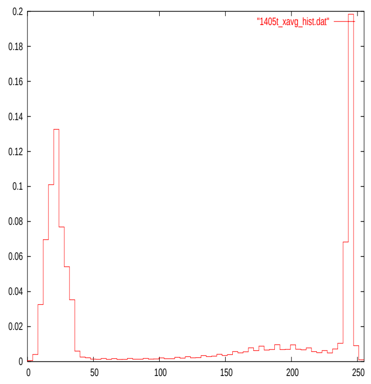
(b)



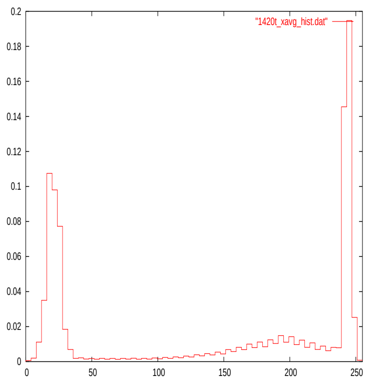
(c)



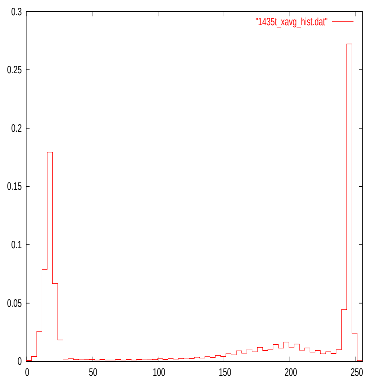
(d)



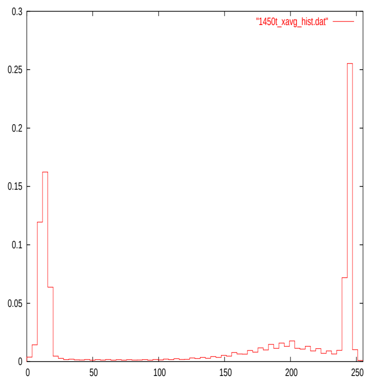
(e)



(f)



(g)



(h)

Figure 11: The probability distribution histograms of the mixing regions in the simulation images at the later time steps. These were obtained using the average along the x-direction for the mixing layer. The number of bins in the histogram is 64. Panels (a) through (h) are time steps 5 through 12.

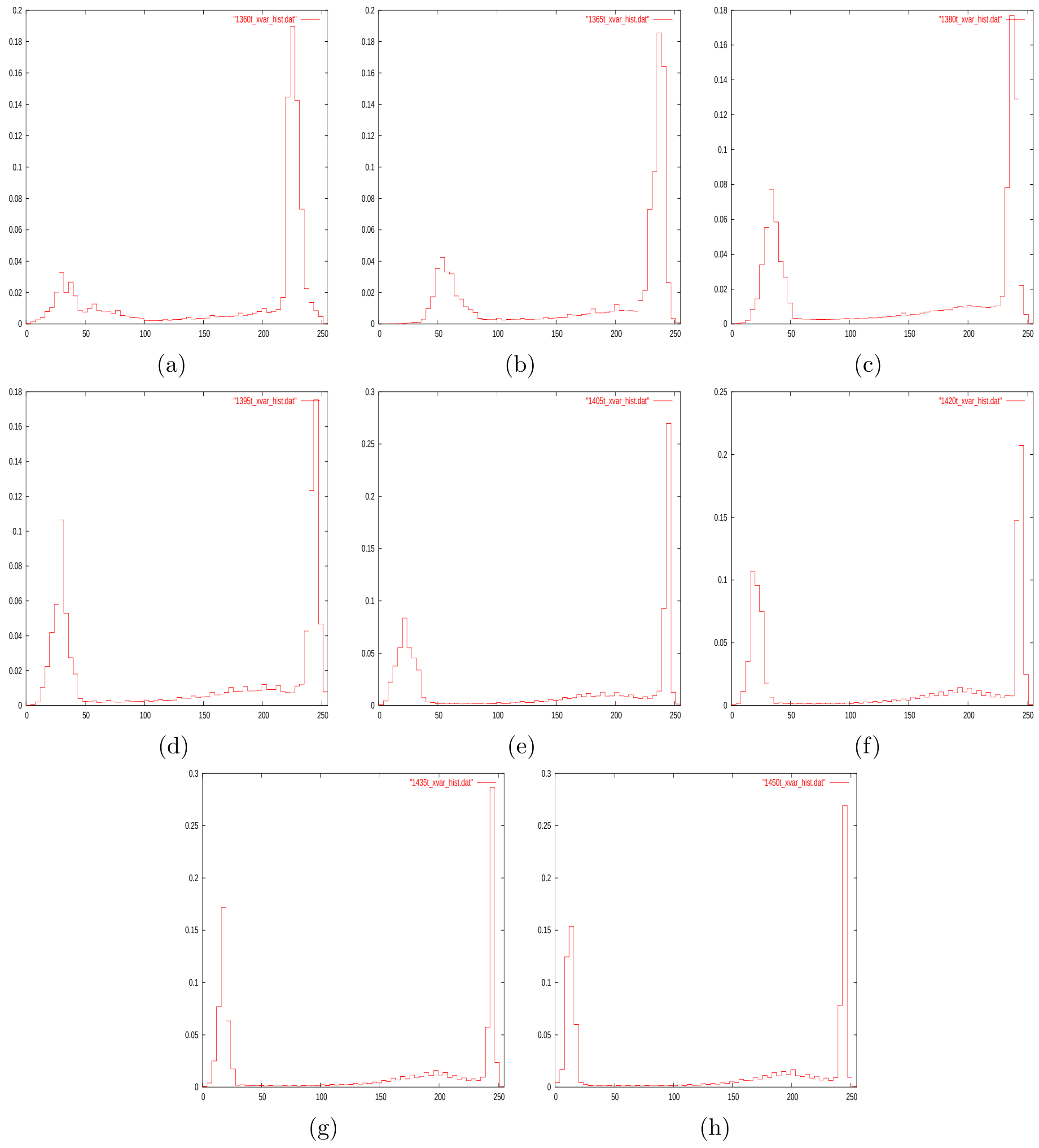


Figure 12: The probability distribution histograms of the mixing regions in the simulation images at the later time steps. These were obtained using the variance along the x-direction for the mixing layer. The number of bins in the histogram is 64. Panels (a) through (h) are time steps 5 through 12.

Appendix A
Processed Images for the Early Time Steps

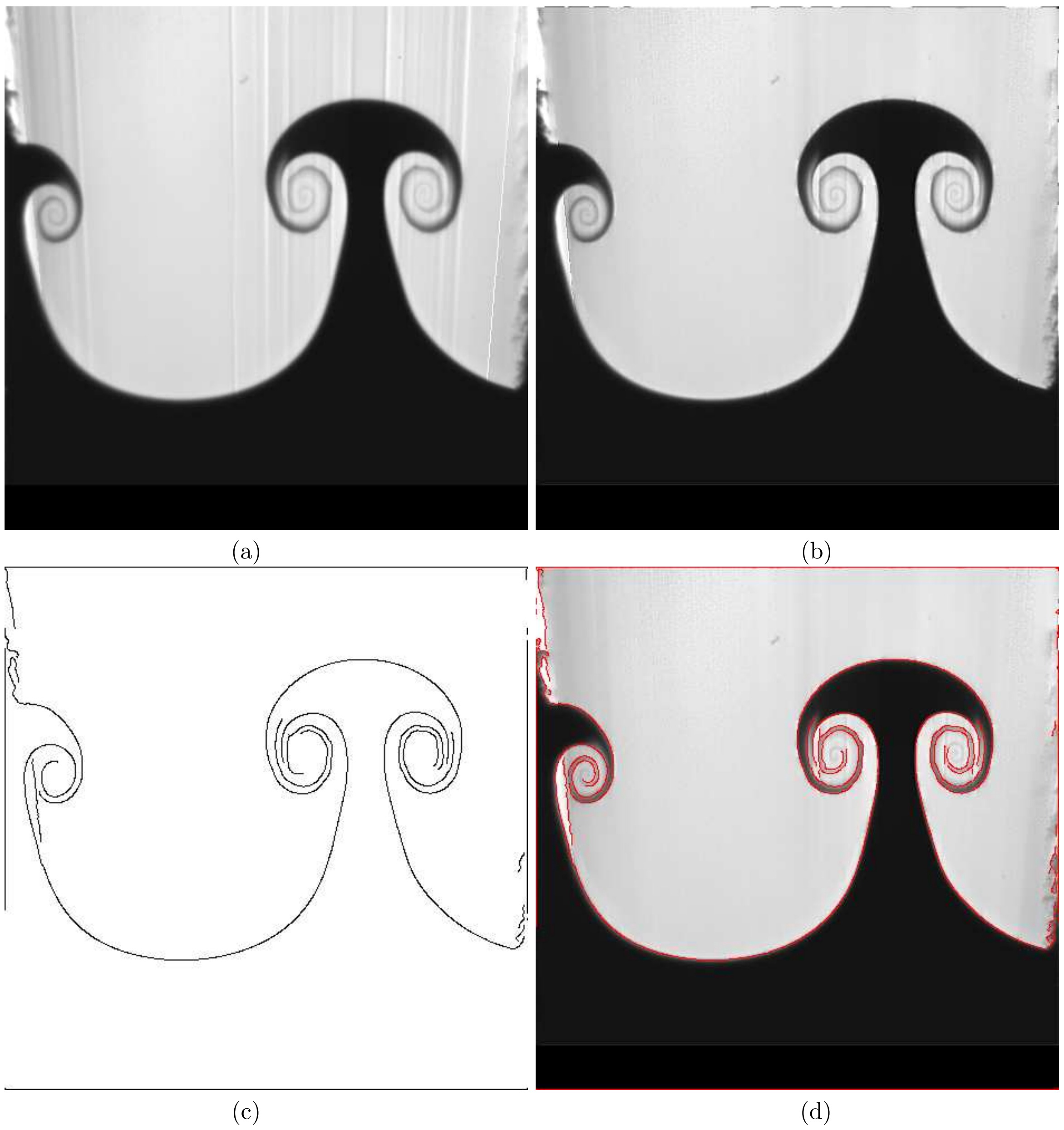


Figure 13: Edge image for experimental data, time step 1: (a) The original noisy experimental image; (b) The cleaned image; (c) The edge image extracted from the cleaned image; (d) The edge image superposed on the cleaned image.

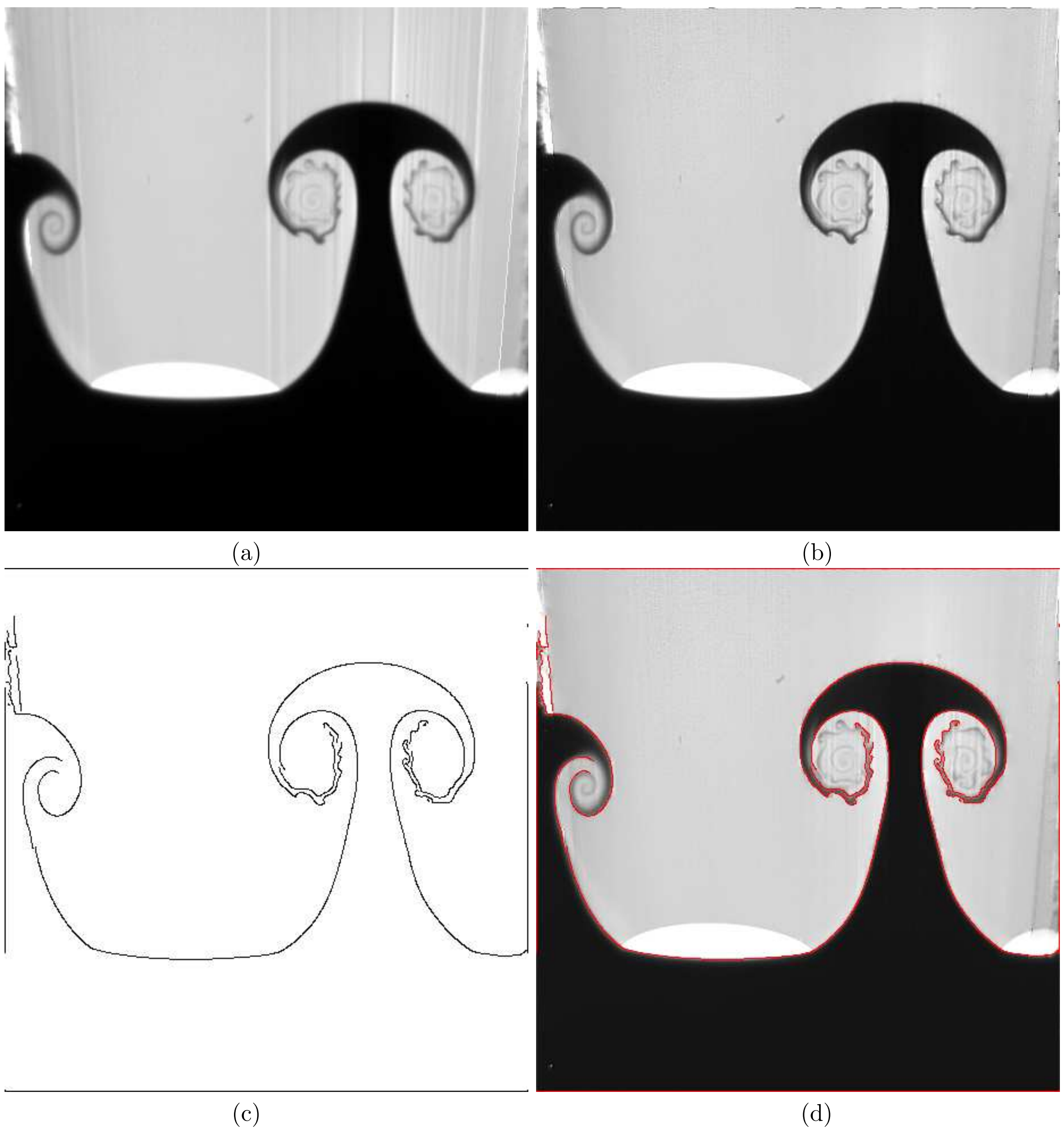


Figure 14: Edge image for experimental data, time step 2: (a) The original noisy experimental image; (b) The cleaned image; (c) The edge image extracted from the cleaned image; (d) The edge image superposed on the cleaned image.

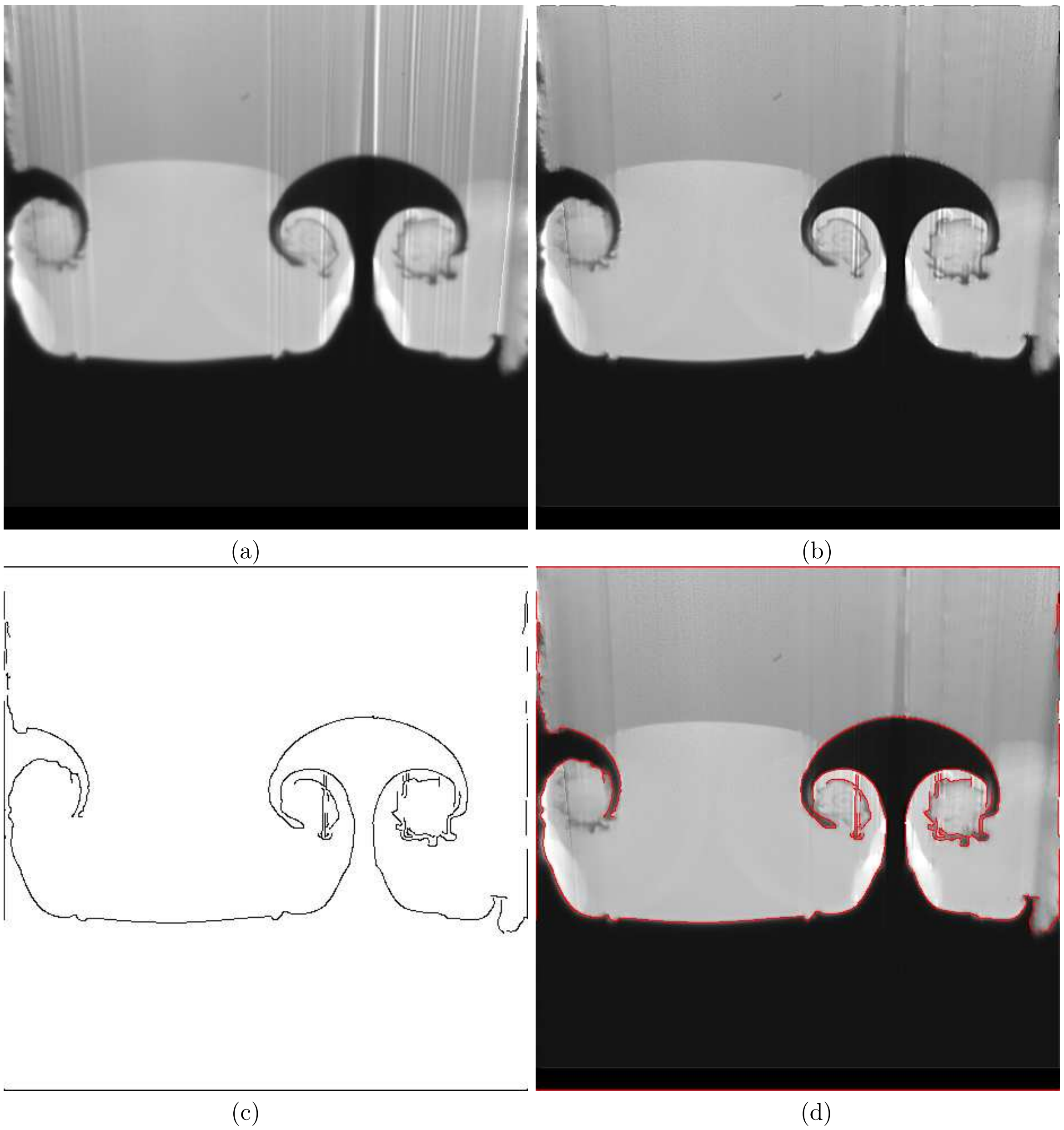
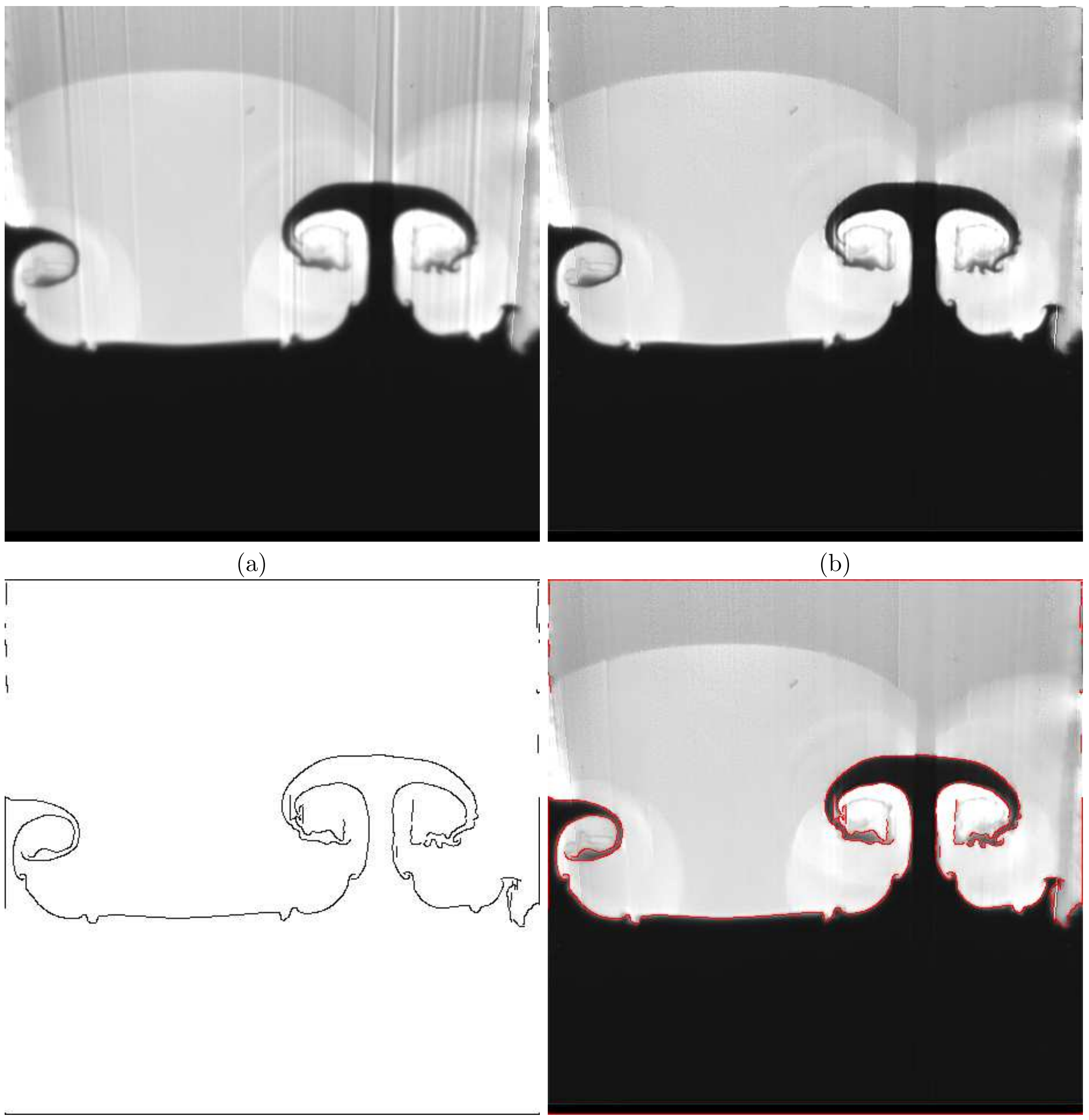
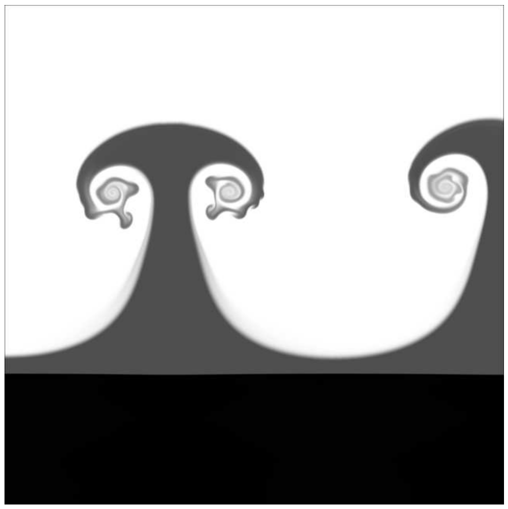
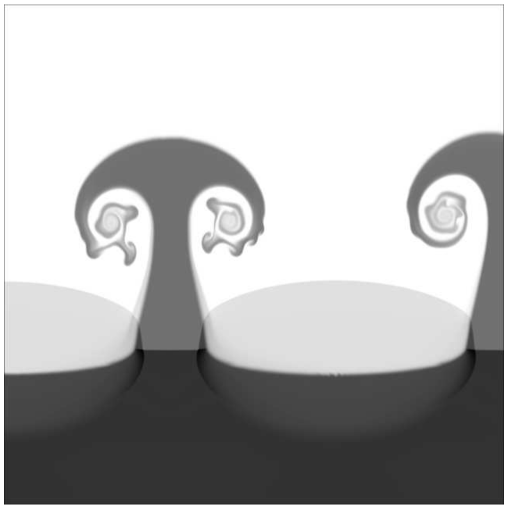


Figure 15: Edge image for experimental data, time step 3: (a) The original noisy experimental image; (b) The cleaned image; (c) The edge image extracted from the cleaned image; (d) The edge image superposed on the cleaned image.

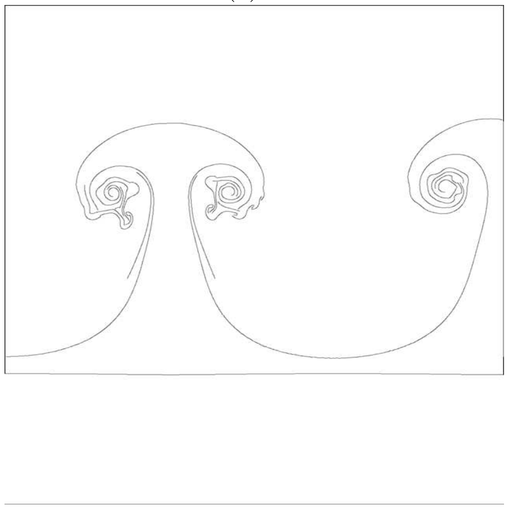
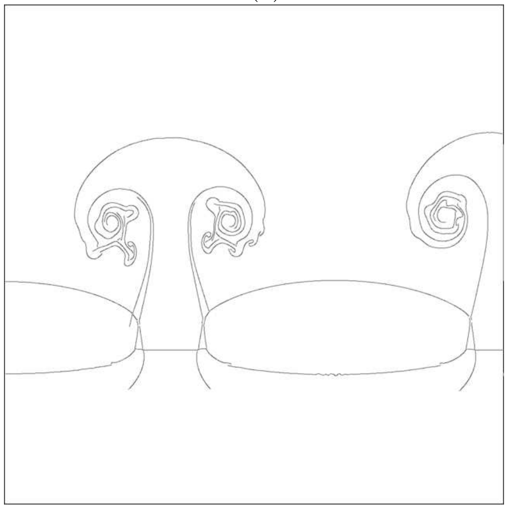


(a) (b) (c) (d)

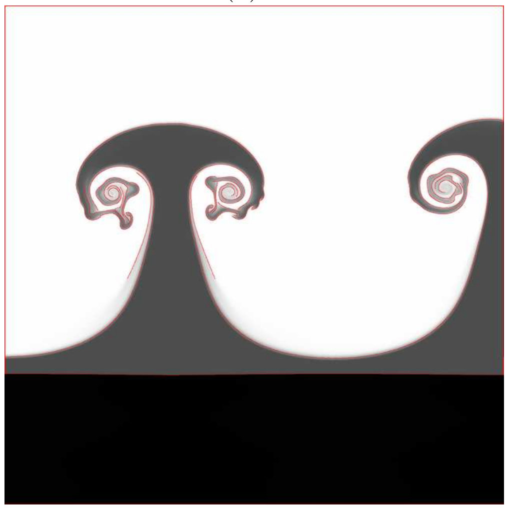
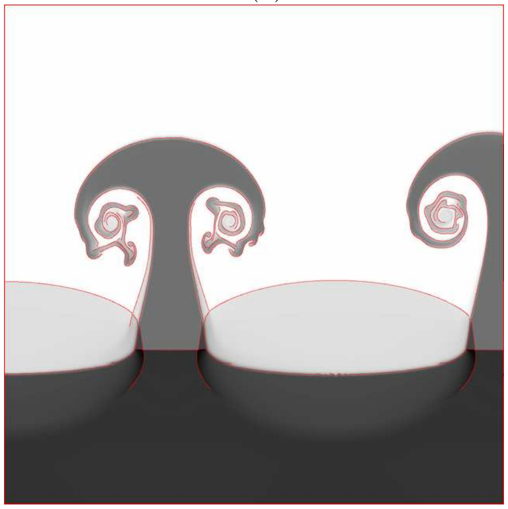
Figure 16: Edge image for experimental data, time step 4: (a) The original noisy experimental image; (b) The cleaned image; (c) The edge image extracted from the cleaned image; (d) The edge image superposed on the cleaned image.

(a) (a)

(b) (b)

(c) (c)

Figure 17: Edge image for simulation data, time step 1 (left column) and time step 2 (right column): (a) The original simulation image; (b) The edge image; (d) The edge image superposed on the original image.

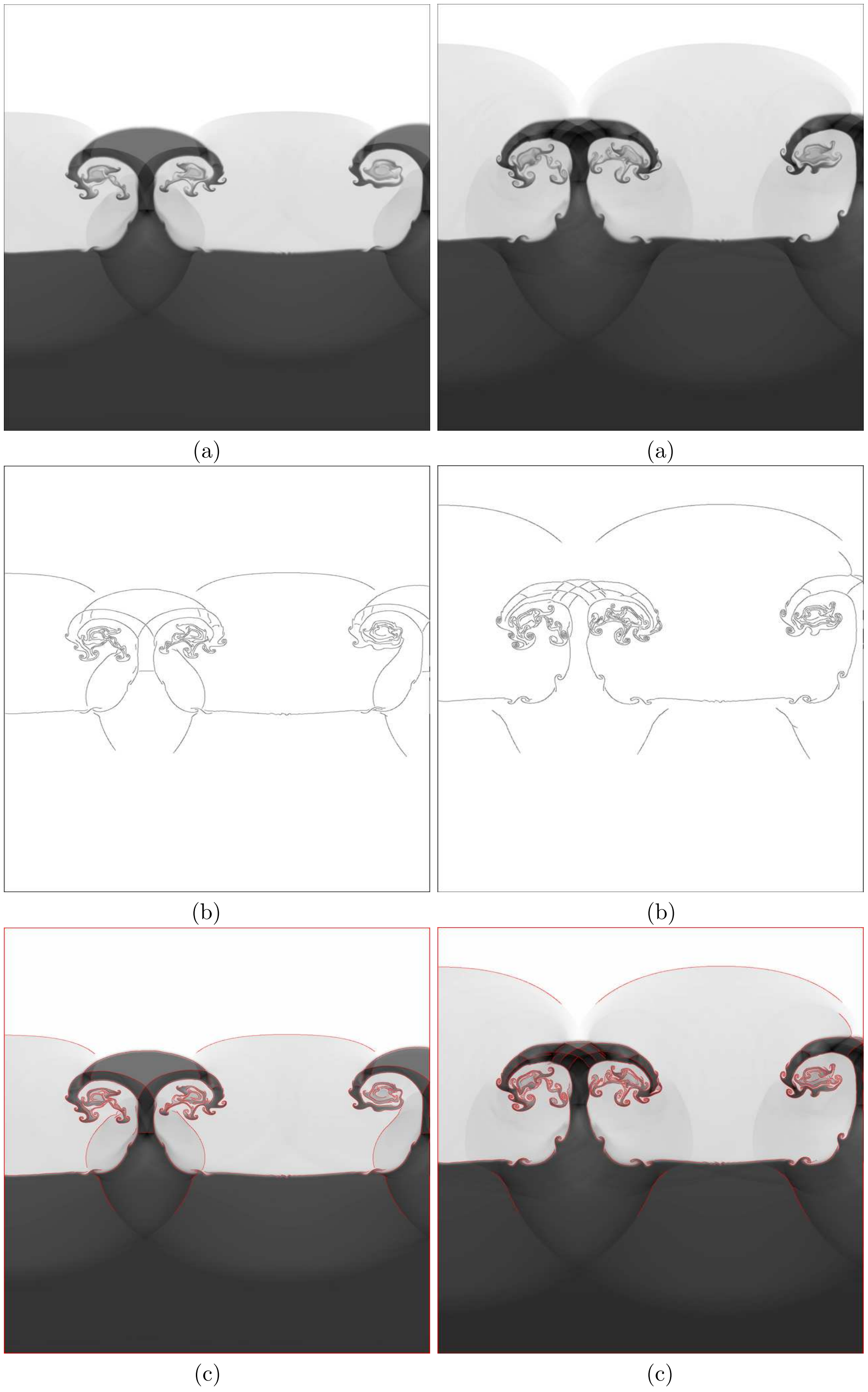


Figure 18: Edge image for simulation data, time step 3 (left column) and time step 4 (right column): (a) The original simulation image; (b) The edge image; (d) The edge image superposed on the original image.

Appendix B
Processed Images for the Later Time Steps

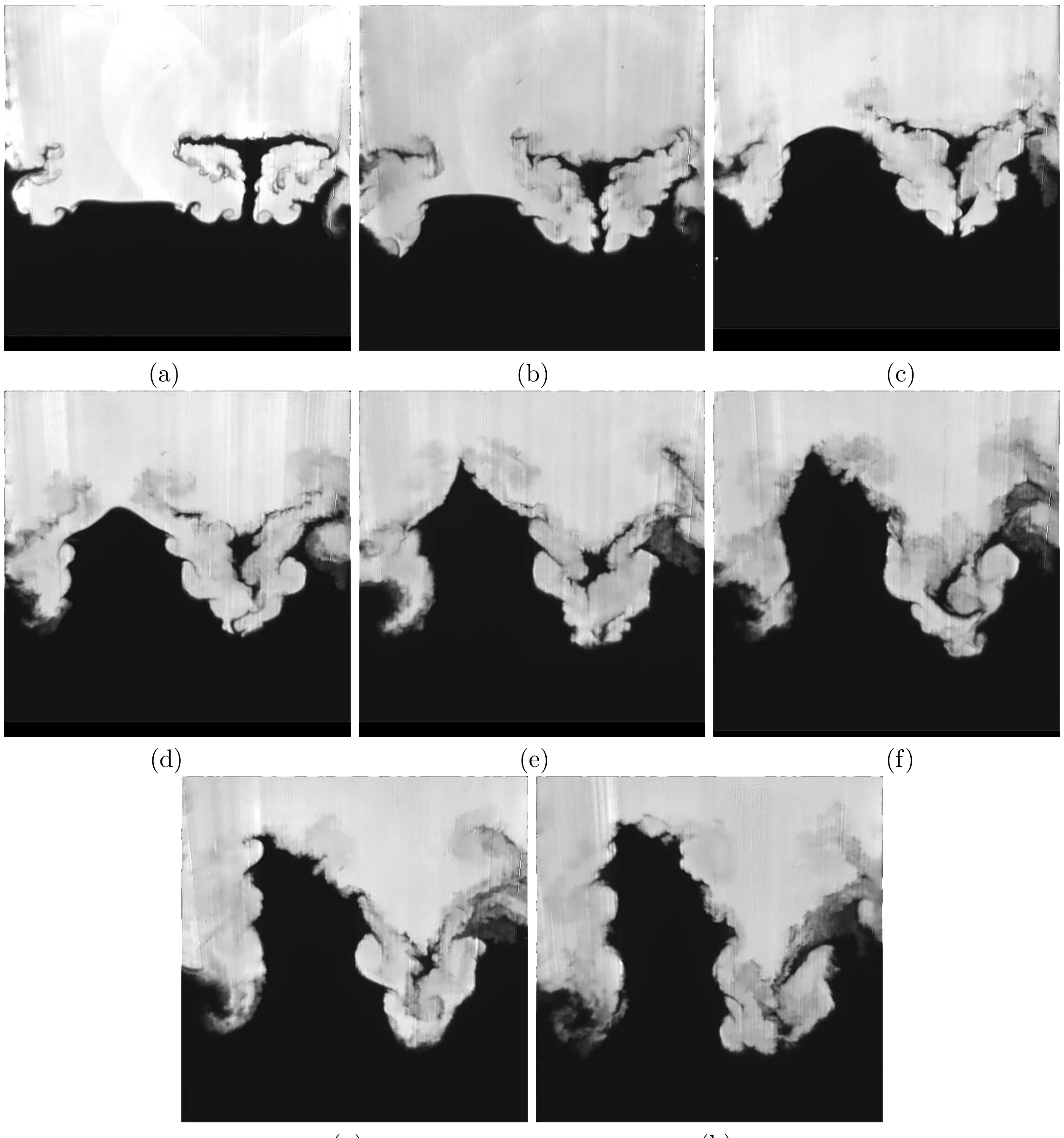
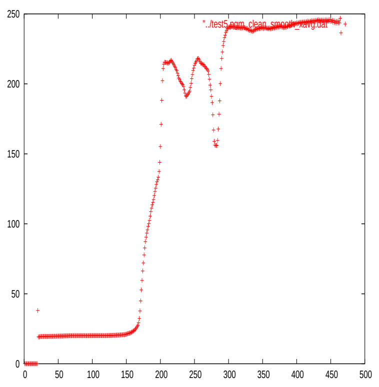
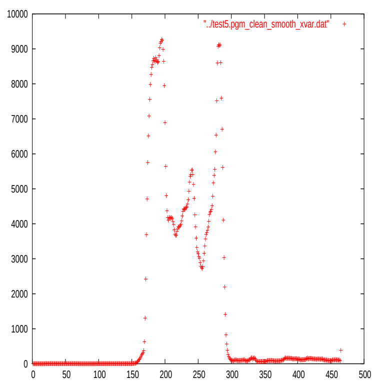

(a) (b) (c) (d) (e) (f) (g) (h)

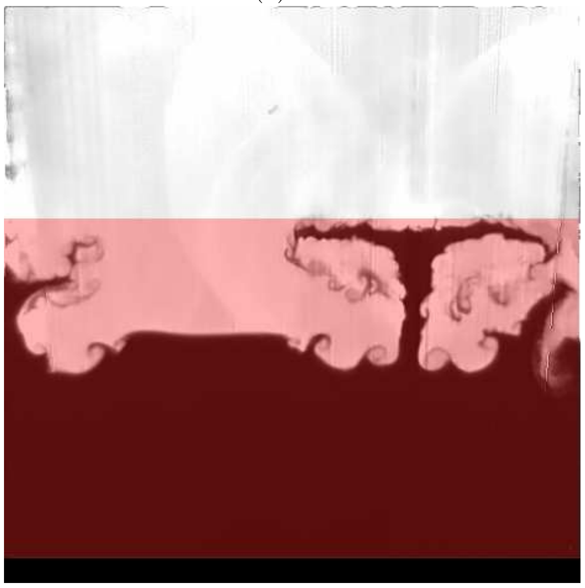
Figure 19: The cleaned experimental images - later time steps.



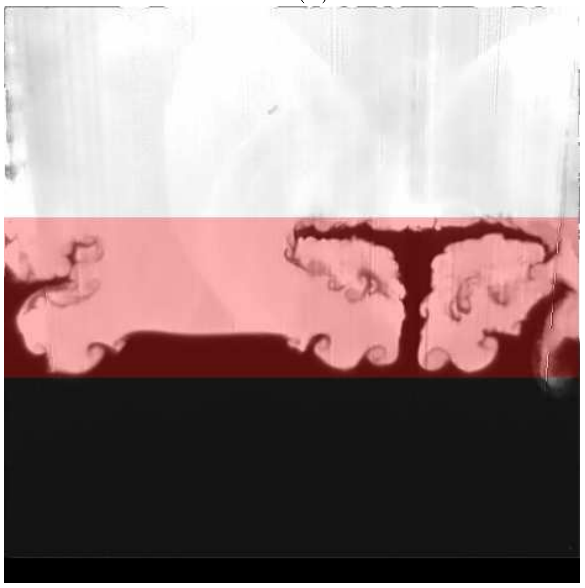
(a)



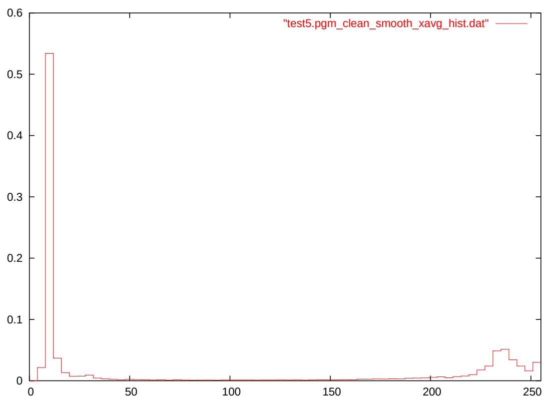
(b)



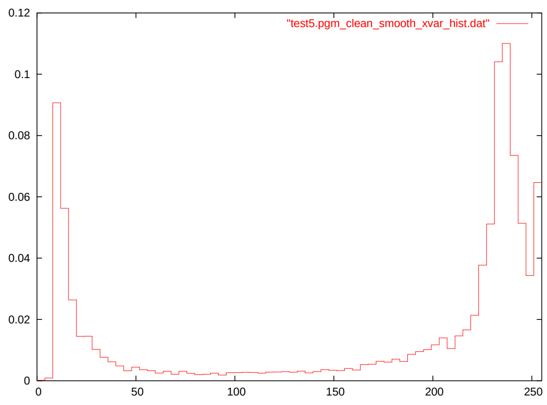
(c)



(d)

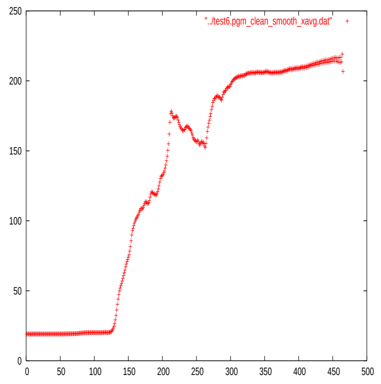


(e)

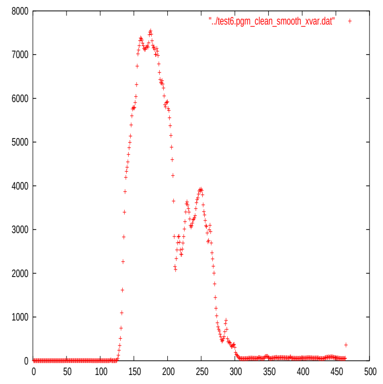


(f)

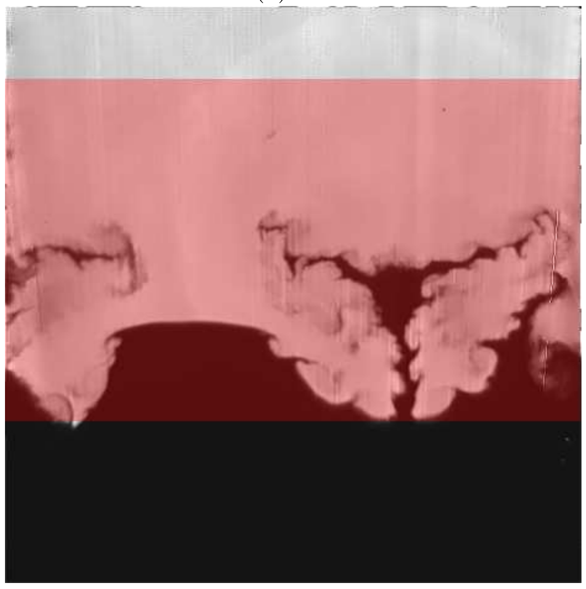
Figure 20: Finding the mixing layer for the experimental data, time step 5, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



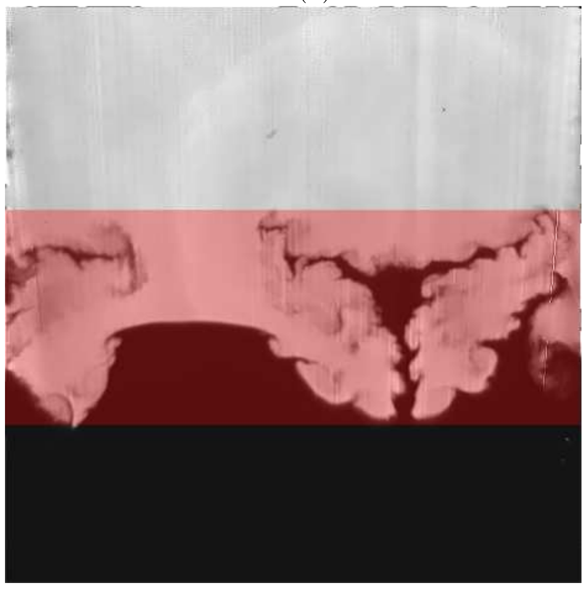
(a)



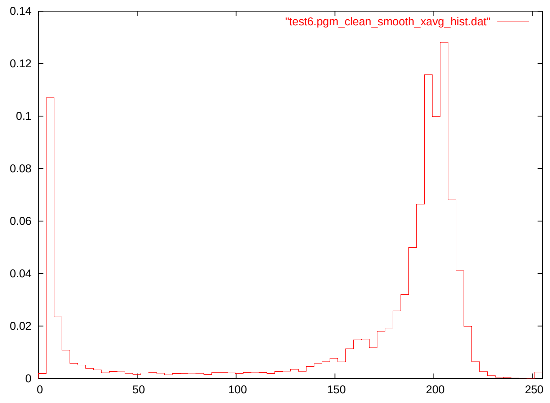
(b)



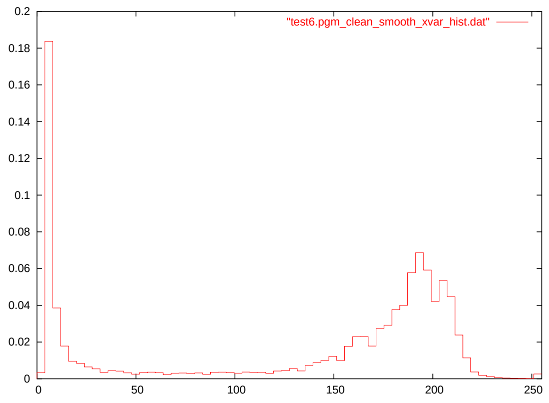
(c)



(d)

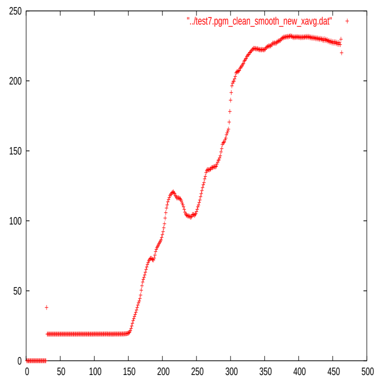


(e)

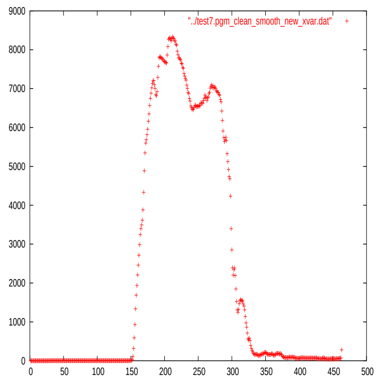


(f)

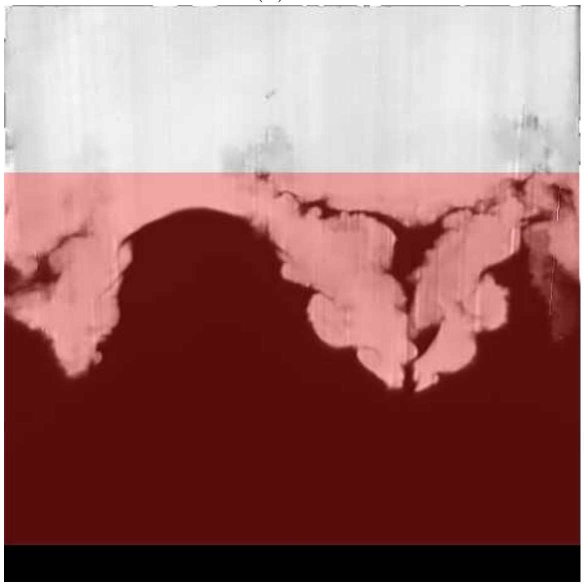
Figure 21: Finding the mixing layer for the experimental data, time step 6, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



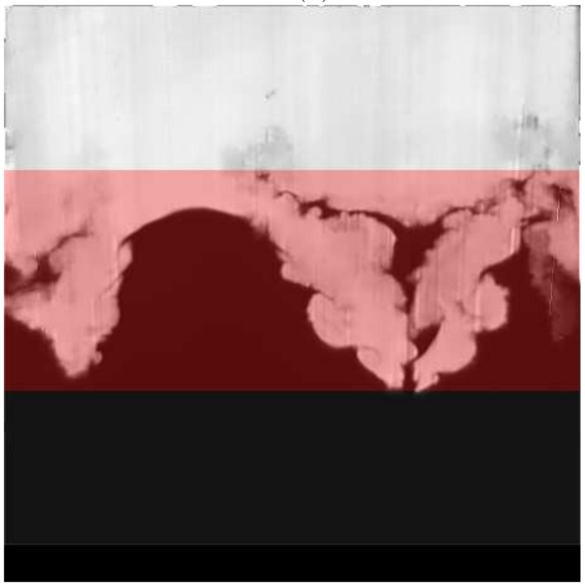
(a)



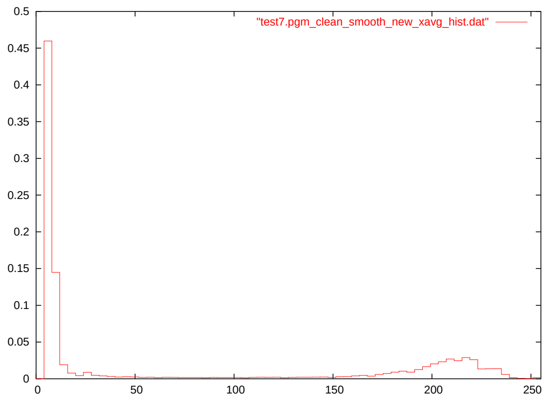
(b)



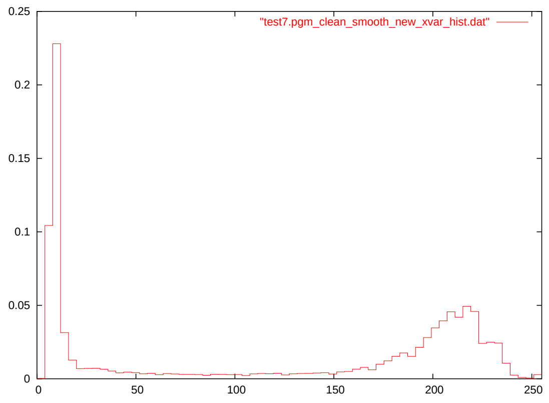
(c)



(d)

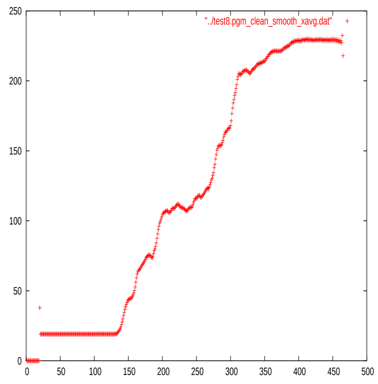


(e)

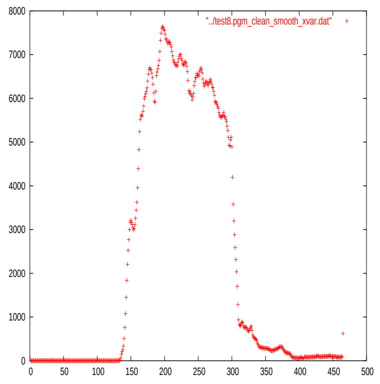


(f)

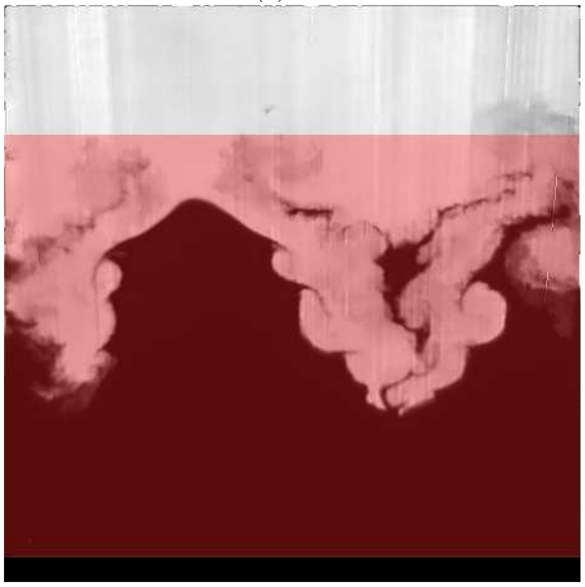
Figure 22: Finding the mixing layer for the experimental data, time step 7, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



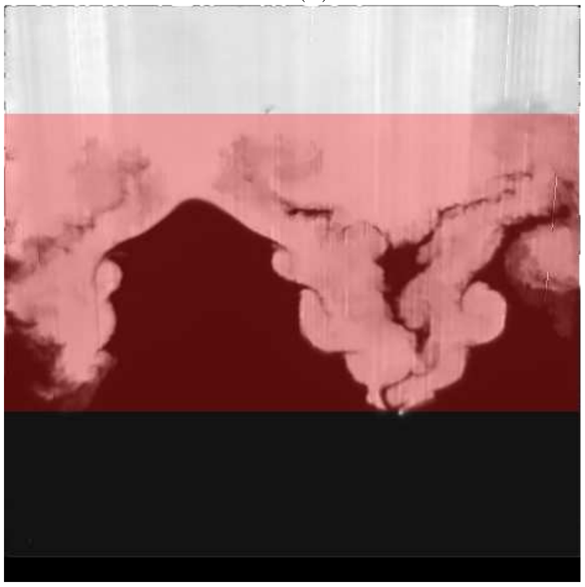
(a)



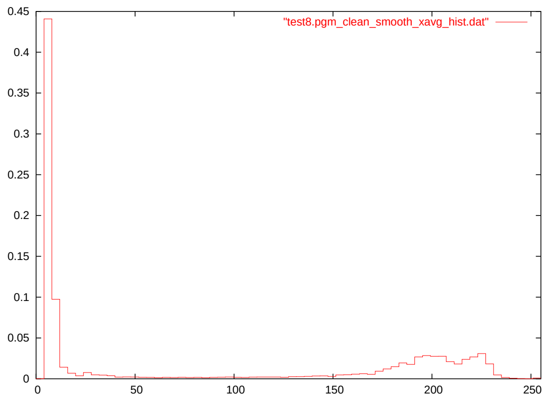
(b)



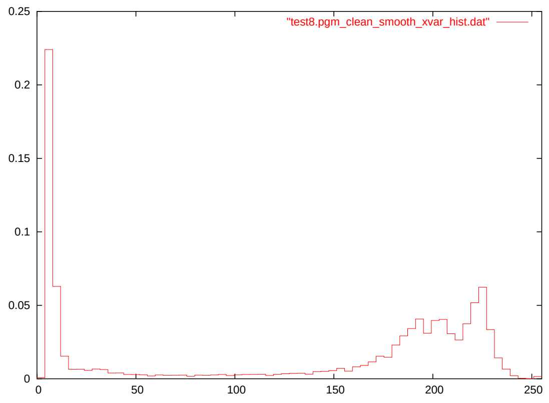
(c)



(d)

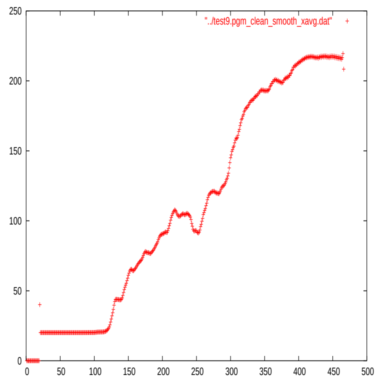


(e)

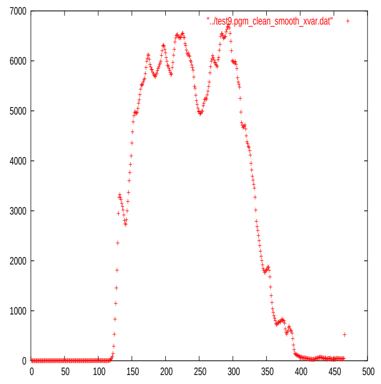


(f)

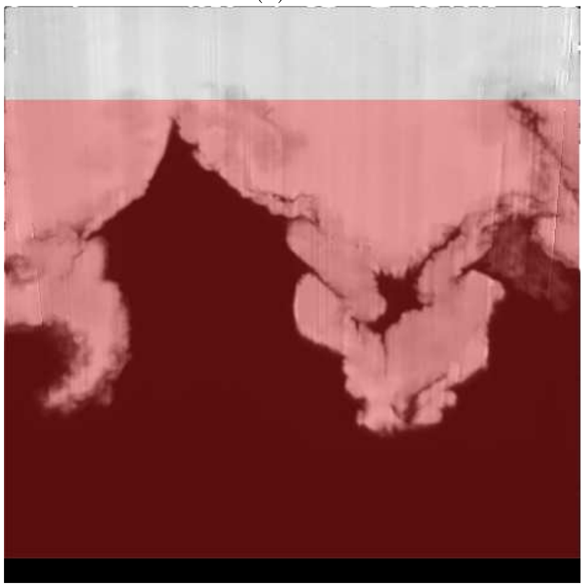
Figure 23: Finding the mixing layer for the experimental data, time step 8, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



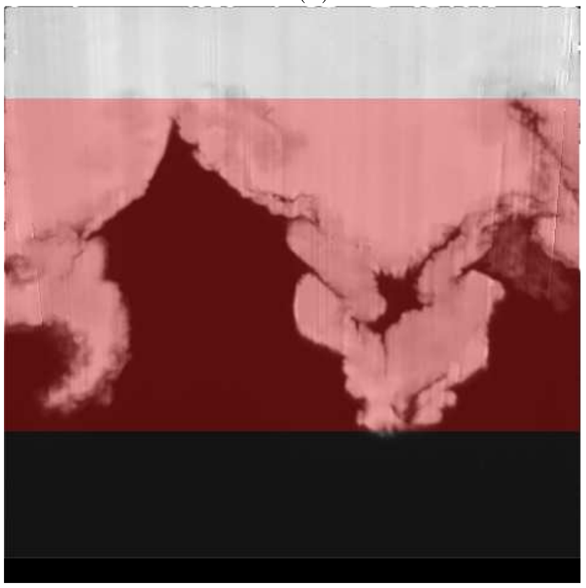
(a)



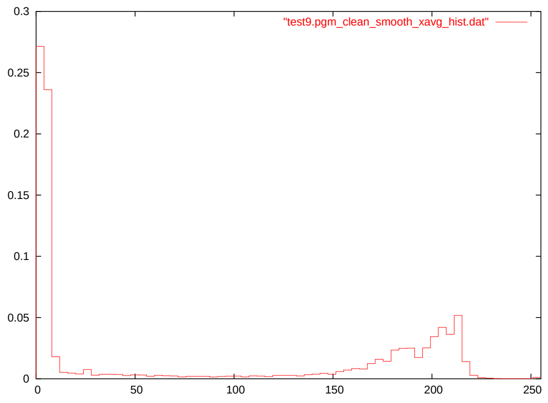
(b)



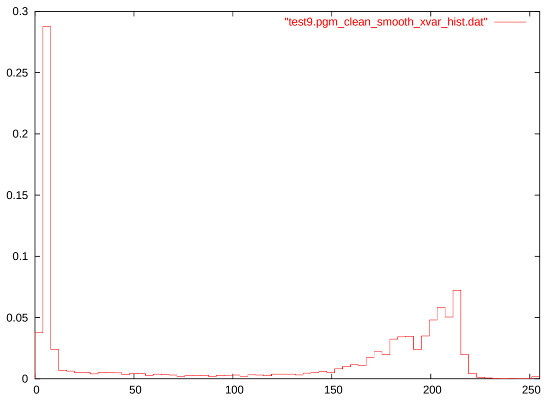
(c)



(d)

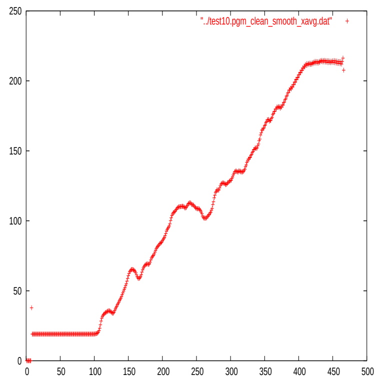


(e)

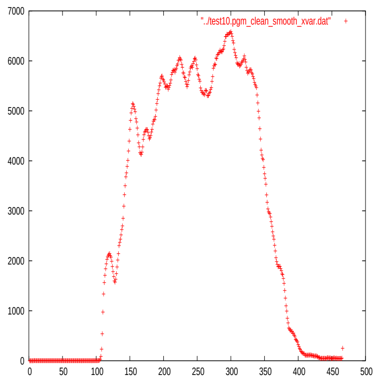


(f)

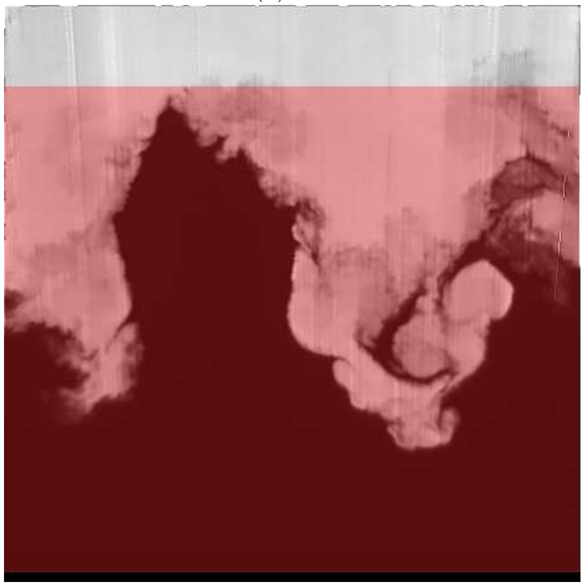
Figure 24: Finding the mixing layer for the experimental data, time step 9, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



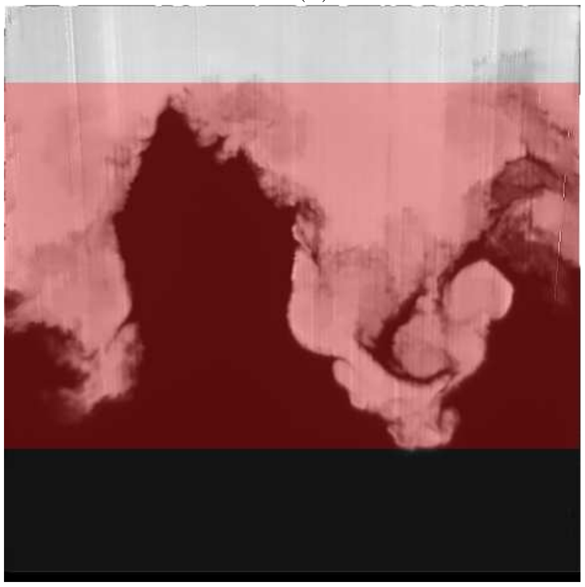
(a)



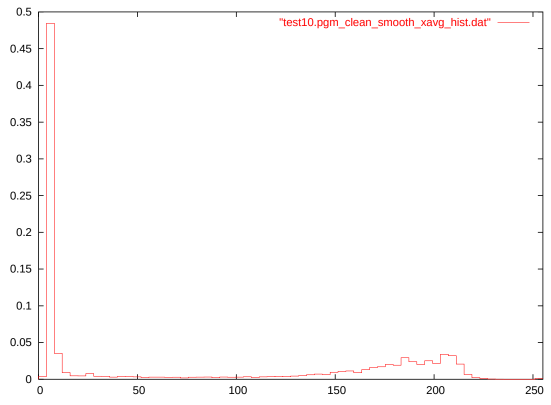
(b)



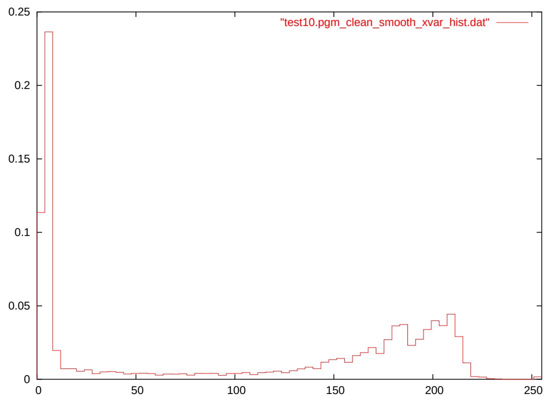
(c)



(d)

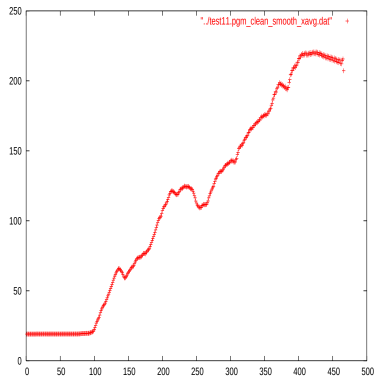


(e)

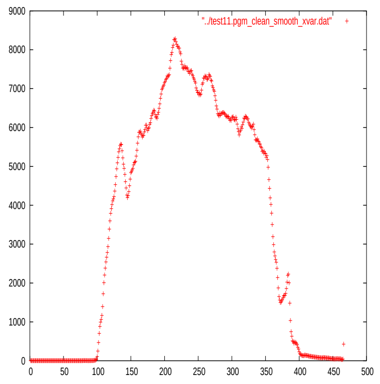


(f)

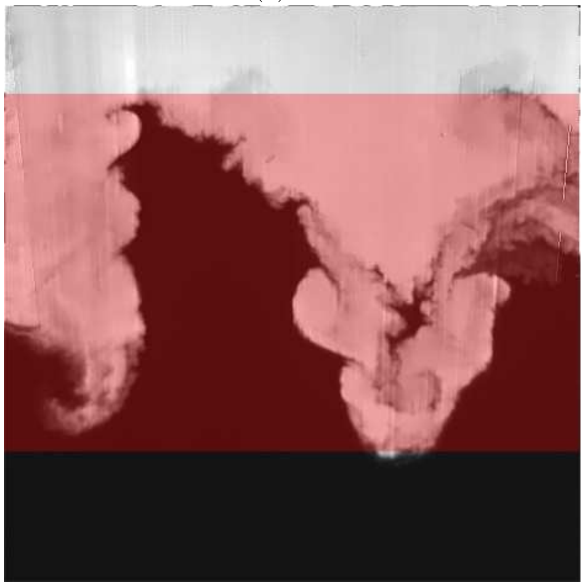
Figure 25: Finding the mixing layer for the experimental data, time step 10, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



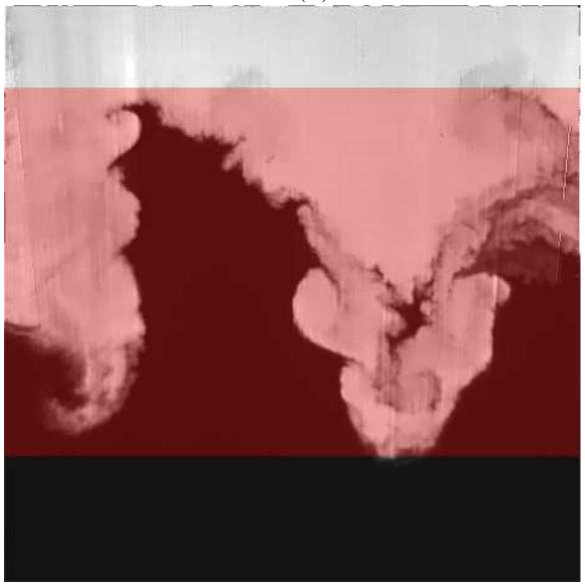
(a)



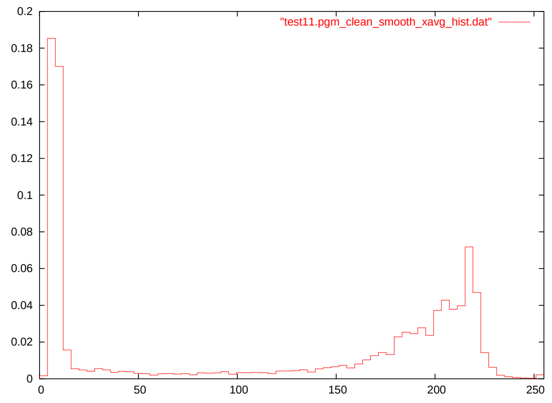
(b)



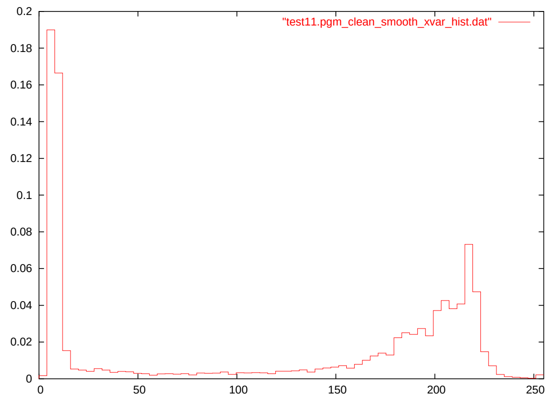
(c)



(d)



(e)



(f)

Figure 26: Finding the mixing layer for the experimental data, time step 11, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).

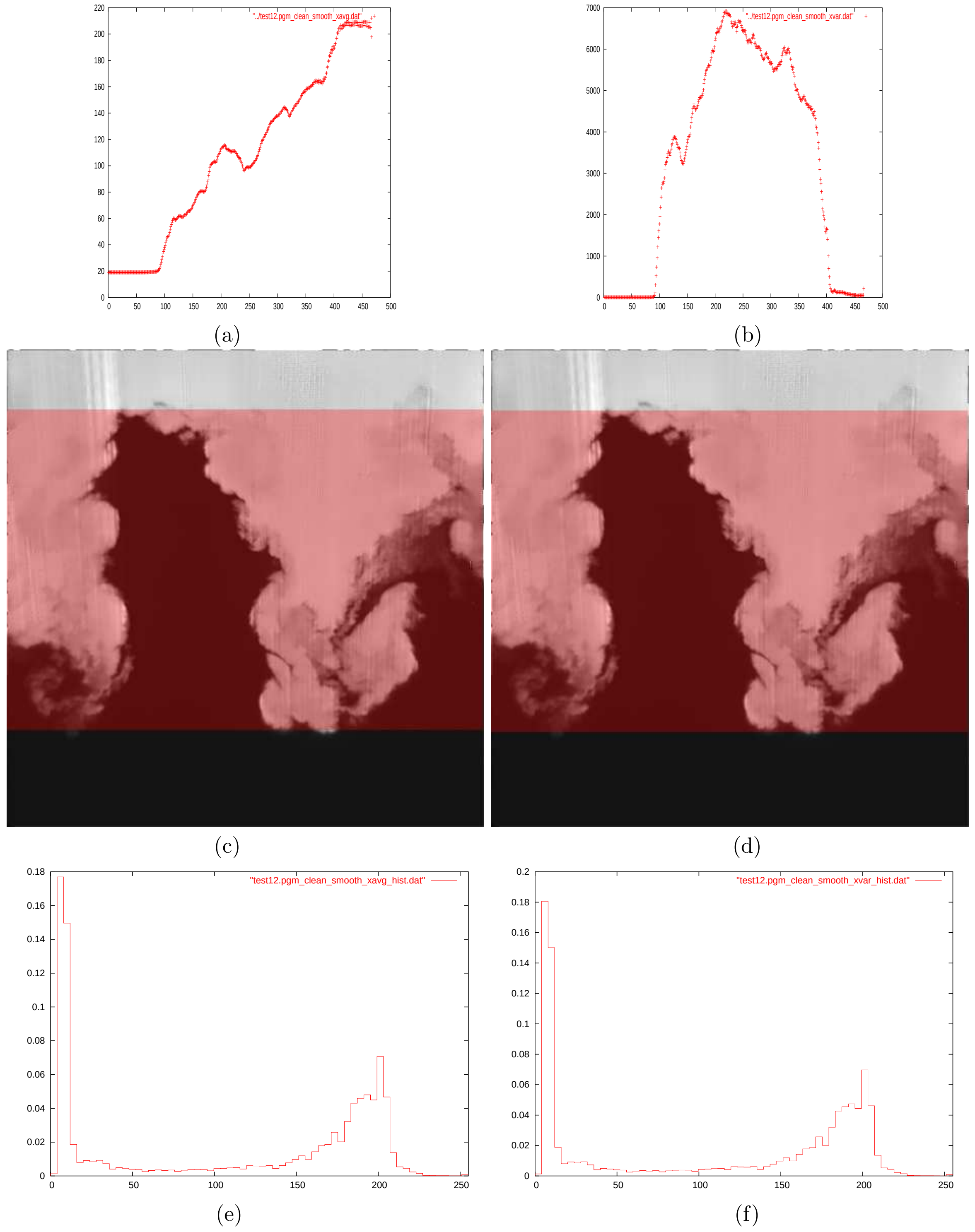
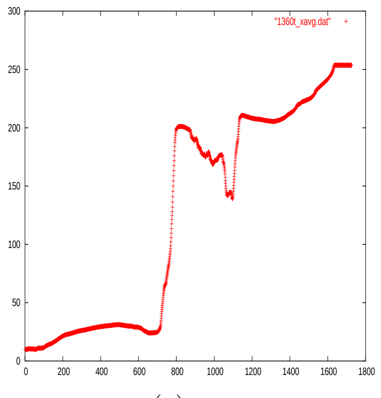
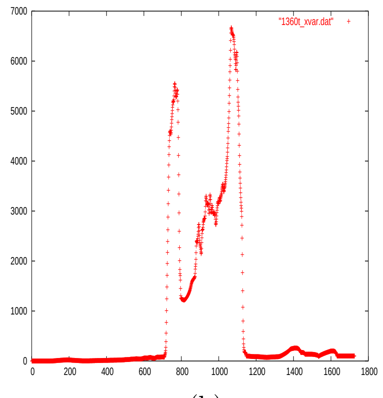


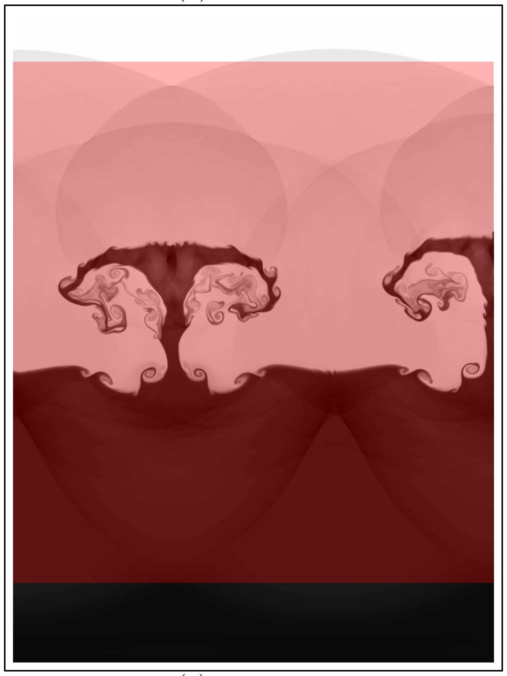
Figure 27: Finding the mixing layer for the experimental data, time step 12, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



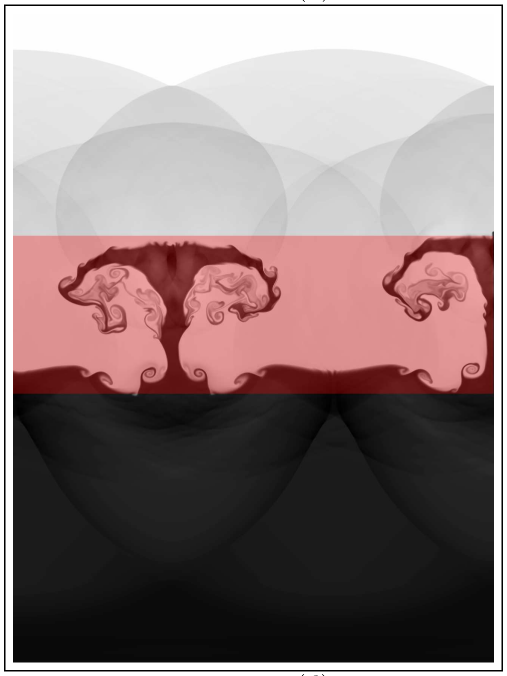
(a)



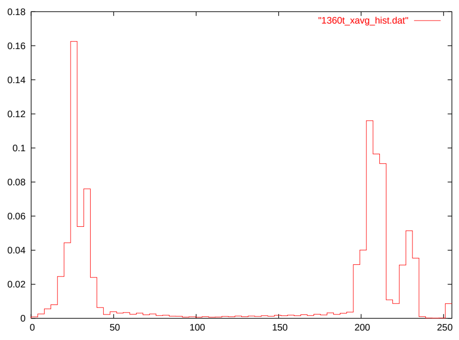
(b)



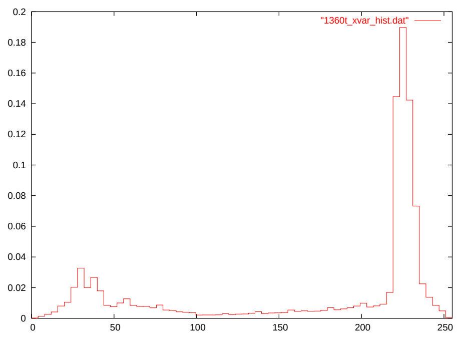
(c)



(d)

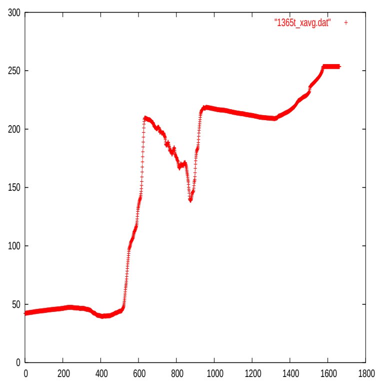


(e)

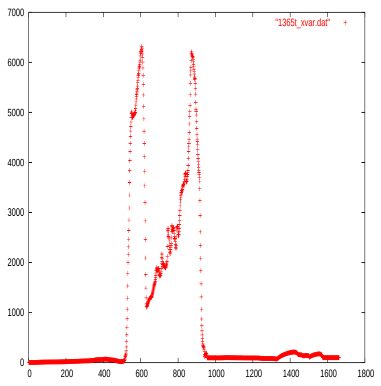


(f)

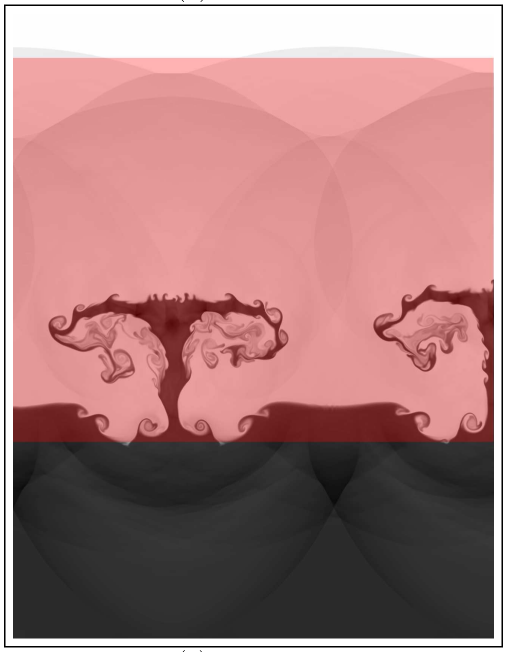
Figure 28: Finding the mixing layer for the simulation data, time step 5, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



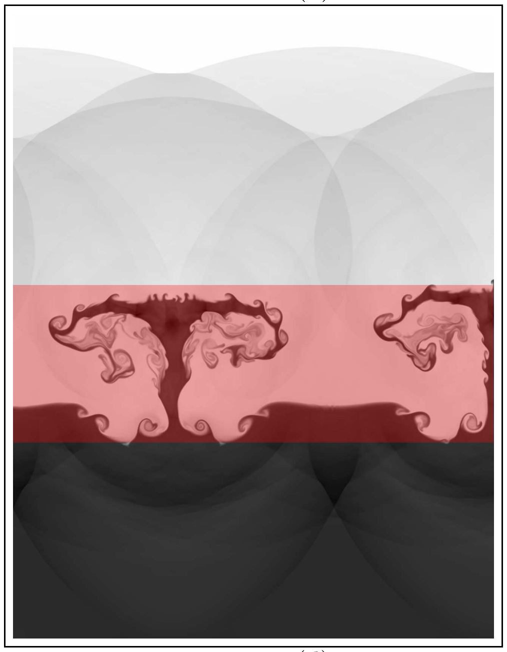
(a)



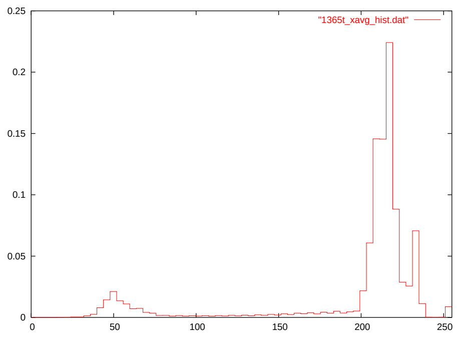
(b)



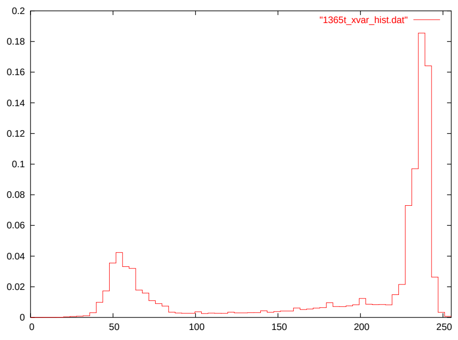
(c)



(d)

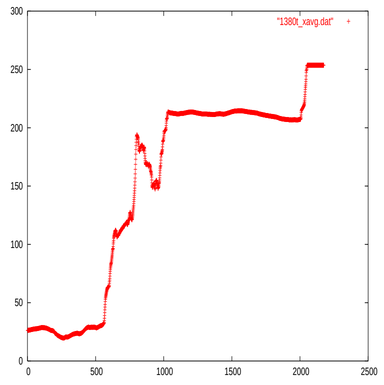


(e)

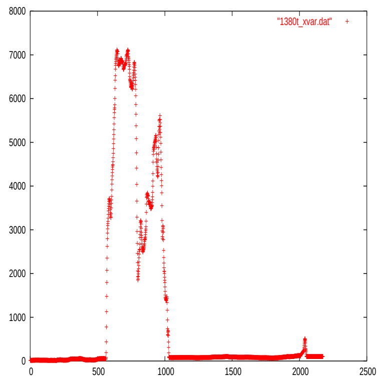


(f)

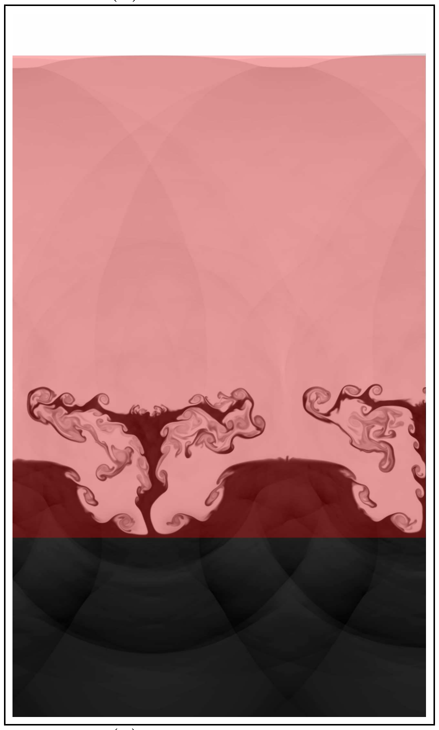
Figure 29: Finding the mixing layer for the simulation data, time step 6, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).

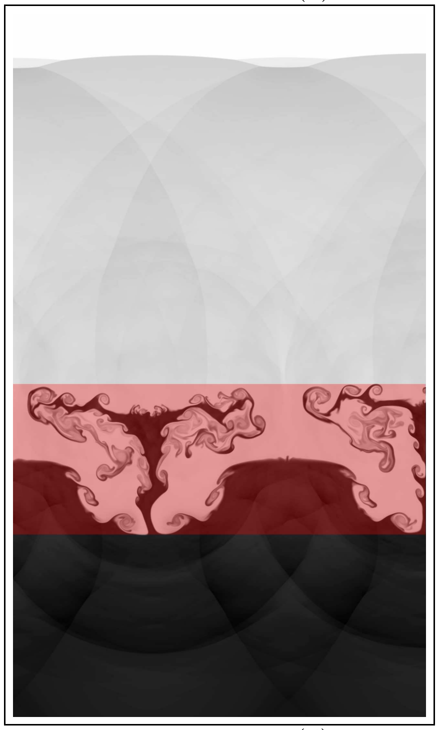


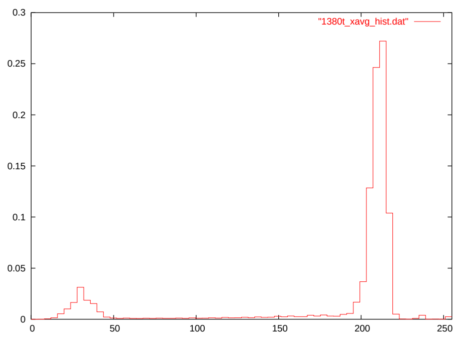
(a)



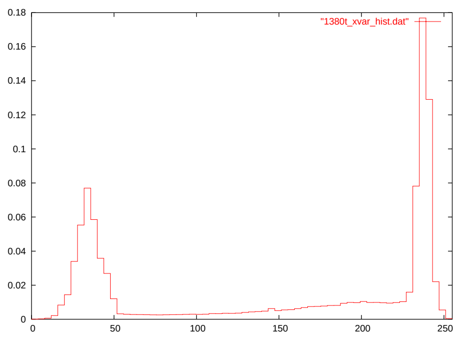
(b)


(c)


(d)



(e)



(f)

Figure 30: Finding the mixing layer for the simulation data, time step 7, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).

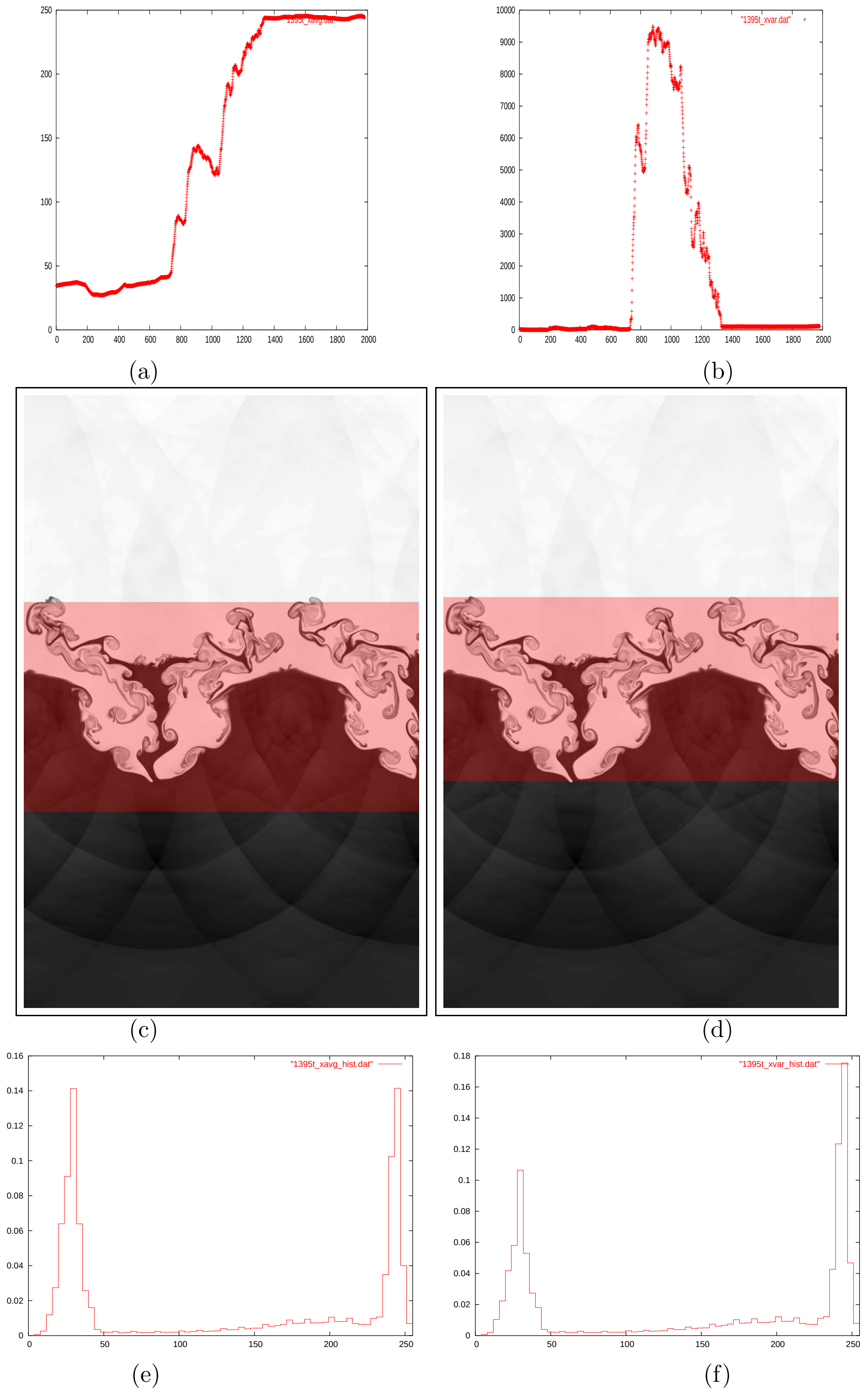
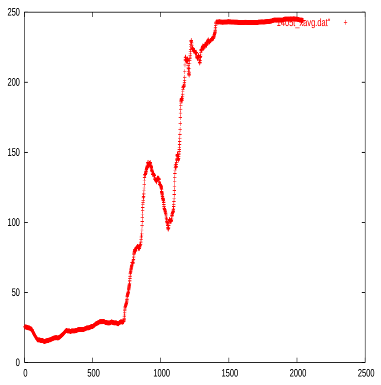
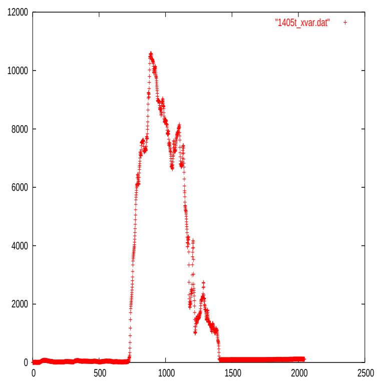


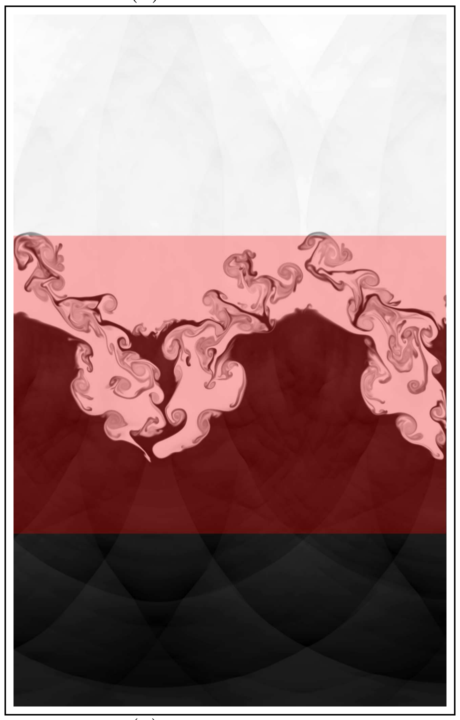
Figure 31: Finding the mixing layer for the simulation data, time step 8, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



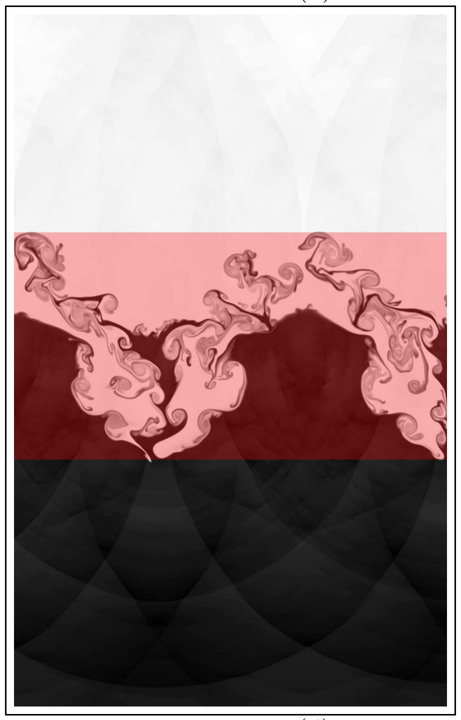
(a)



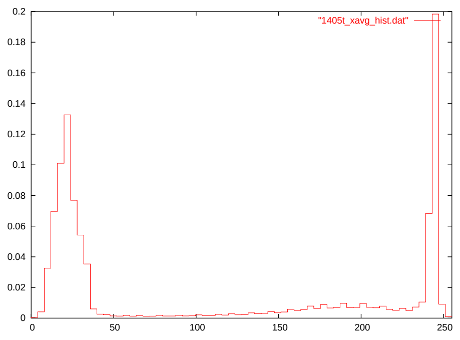
(b)



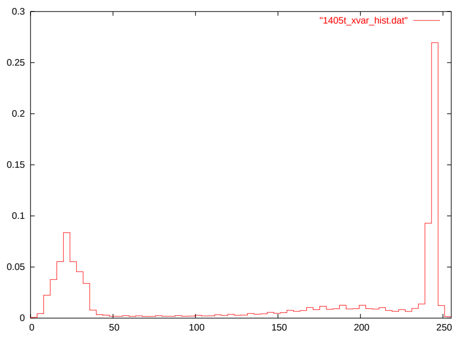
(c)



(d)

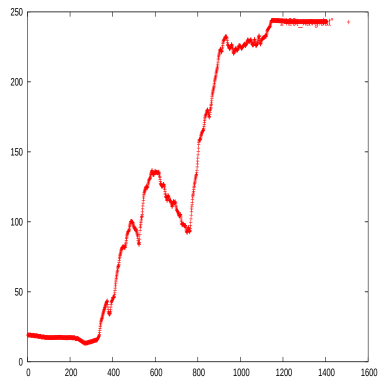


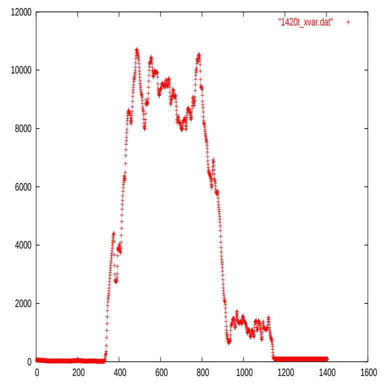
(e)



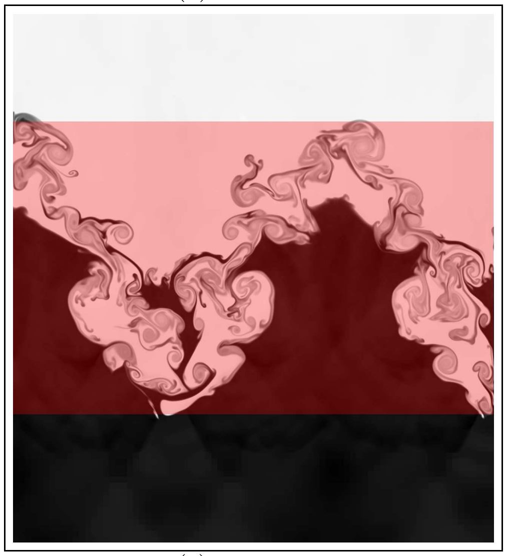
(f)

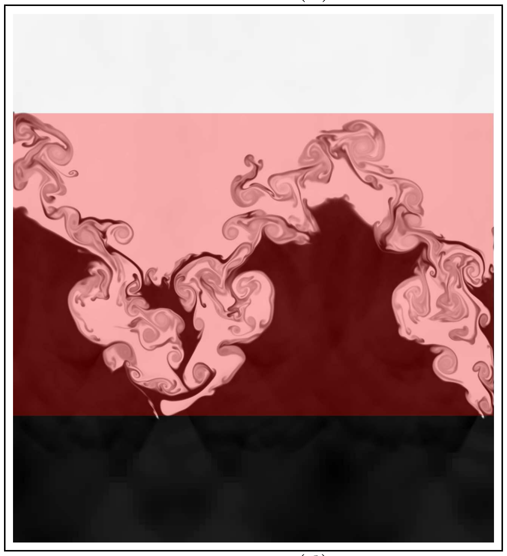
Figure 32: Finding the mixing layer for the simulation data, time step 9, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).

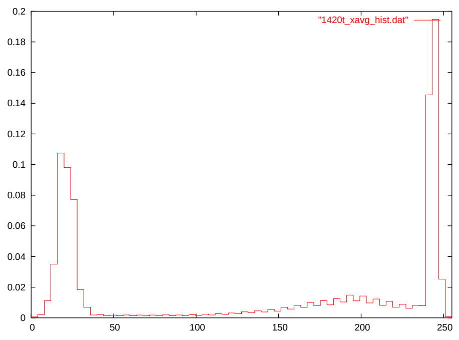

(a)



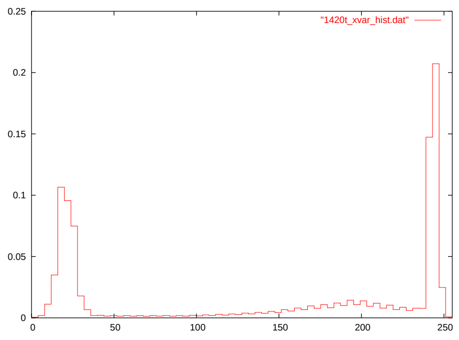
(b)


(c)


(d)

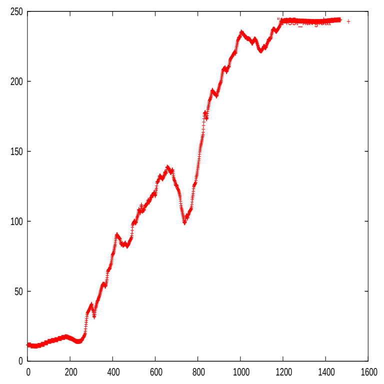


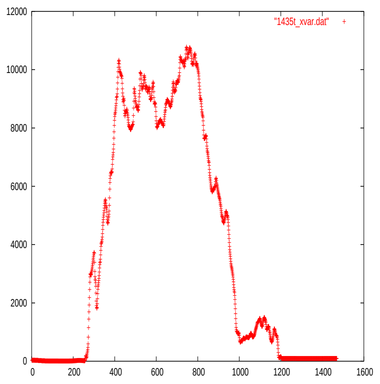
(e)



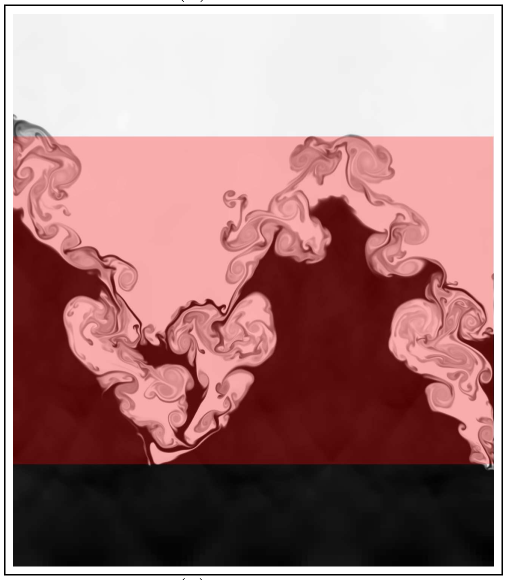
(f)

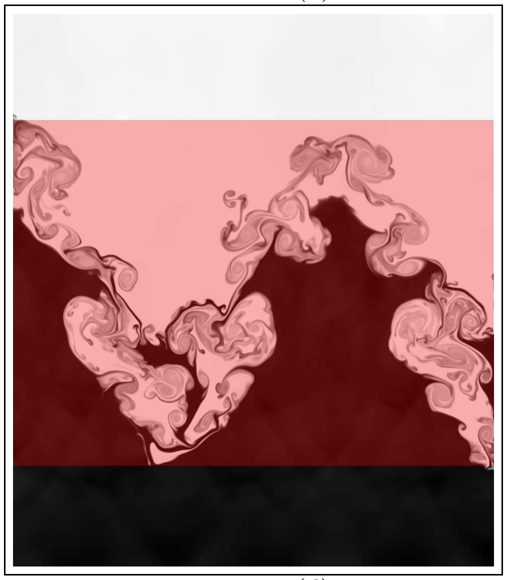
Figure 33: Finding the mixing layer for the simulation data, time step 10, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).

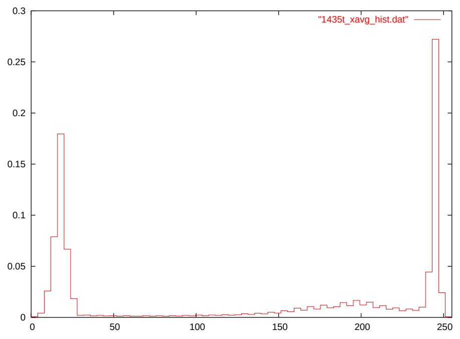

(a)



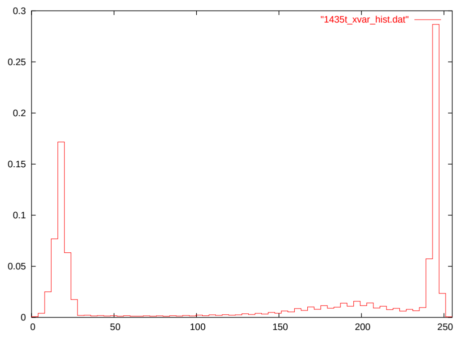
(b)


(c)


(d)

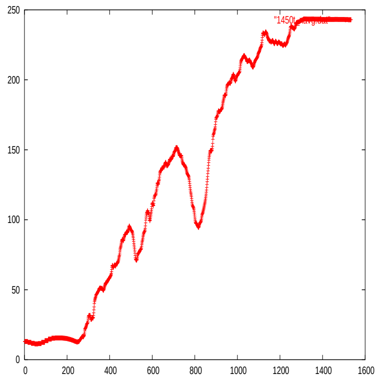


(e)

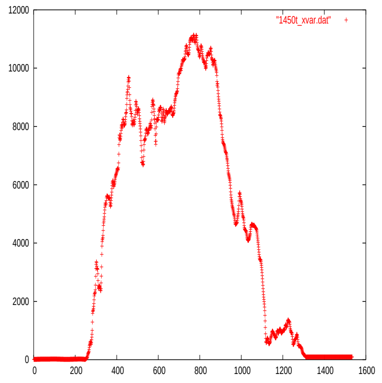


(f)

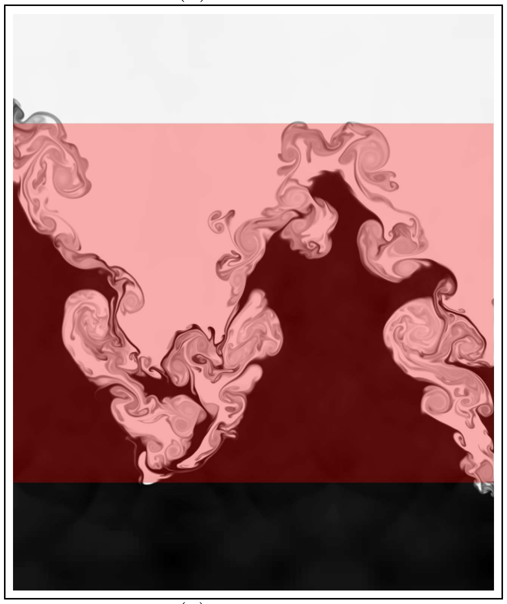
Figure 34: Finding the mixing layer for the simulation data, time step 11, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).



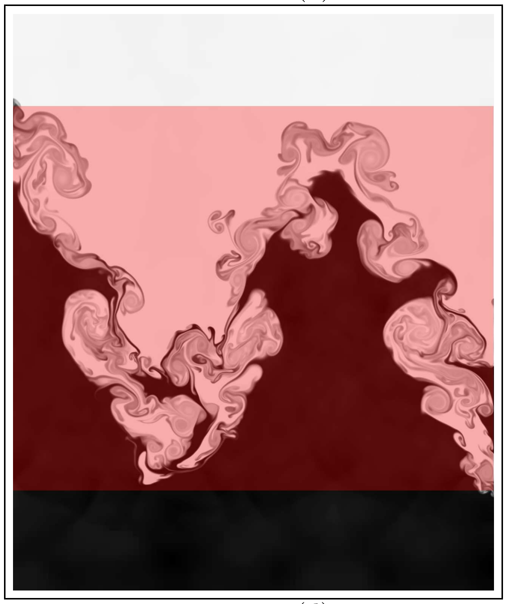
(a)



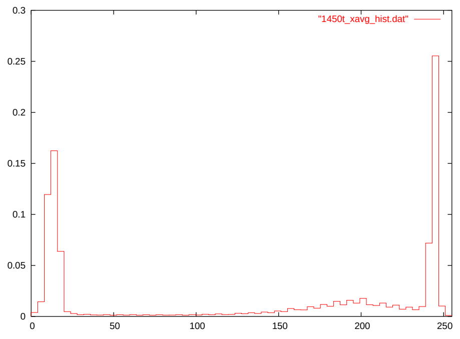
(b)



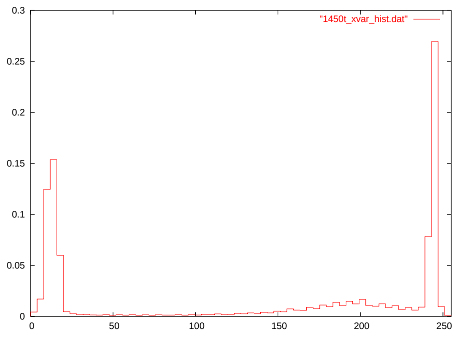
(c)



(d)



(e)



(f)

Figure 35: Finding the mixing layer for the simulation data, time step 12, using the average-based (left column) and the variance-based (right column) approach. (a) (and (b)) The average (variance) of pixel intensities in the x direction. (c) (and (d)) The height of the mixing layer (highlighted) obtained using the average (variance). (e) (and (f)) The 64-bin PDF of the mixing layer obtained using the average (variance).