IMAGE ANALYSIS FOR VALIDATION OF SIMULATIONS OF A FLUID MIX PROBLEM

Chandrika Kamath and Paul L. Miller

Lawrence Livermore National Laboratory
Livermore, CA 94551

ABSTRACT

As computer simulations gain acceptance for the modeling of complex physical phenomena, there is an increasing need to validate these simulation codes by comparing them to experiments. Currently, this is done qualitatively, using a visual approach. This is obviously very subjective and more quantitative metrics are needed, especially to identify simulations which are closer to experiments than other simulations. In this paper, we show how image processing techniques can be effectively used in such comparisons. Using an example from the problem of mixing of two fluids, we show that we can quantitatively compare experimental and simulation images by extracting higher level features to characterize the objects in the images.

Index Terms— Code validation, experimental images, denoising, feature extraction, image analysis

1. INTRODUCTION

Computer simulations enable us to understand complex phenomena through the analysis of mathematical models on high performance computers. By filling the gap between physical experiments and analytical approaches, computer simulations provide both qualitative and quantitative insights into physical phenomena. They are particularly useful when the phenomena is too complex to be analyzed analytically, such as the flow around an airplane, or too expensive for extensive experimentation, such as car crash tests.

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 or to refine the mathematical model underlying the simulation.

Typically, the image from a computer simulation is compared visually to an experimental image or an image from another simulation. While such a qualitative comparison provides a coarse metric, there is an increasing need for more quantitative and objective metrics. This can be a challenge as the simulation and experimental images may be of very different sizes and appear quite different. Further, the experimental images can be quite noisy, with a low contrast in the regions of interest. Often, scientists are not interested in comparing the entire simulation image to the experimental image, but in comparing certain features of the objects in the two images. In this paper, we describe how image processing techniques can be used to provide a quantitative comparison of simulation and experimental images in the context of a problem of mixing of two fluids.

2. PROBLEM DESCRIPTION

We consider the problem of Richtmyer-Meshkov instability [1] 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. Scientists are interested in understanding what happens to this instability over time, that is, how do the two fluids behave once the shock wave hits the interface between them.

In our analysis, we focus on the problem where the interface between a column of acetone/air mixture on the top of a shock tube and a column of sulphur hexachloride at the bottom, is perturbed by a shock wave at Mach 1.3 [2]. The goal is to investigate what happens to the interface when the shock wave, having gone through the interface once, is reflected off the lower wall of the shock tube.

The experimental images for this problem are obtained using planar laser-induced fluorescence (PLIF) imagery, while the simulation data is from a multi-material Eulerian adaptive mesh refinement code. Figure 1 shows the data for the experiments (top row) and the simulations (bottom row) as the two fluids mix over time. The left panel shows the two fluids (in grey and in black) after the shock has first passed through. Note the $1\frac{1}{2}$ mushroom-shaped structure in the images. In Panel (b), the shock wave reflected off the lower wall can be seen as the white (grey) region between the two mushrooms in the experiments (simulations). In panels (c) through (e), the re-shock travels further and the mushroom structures start
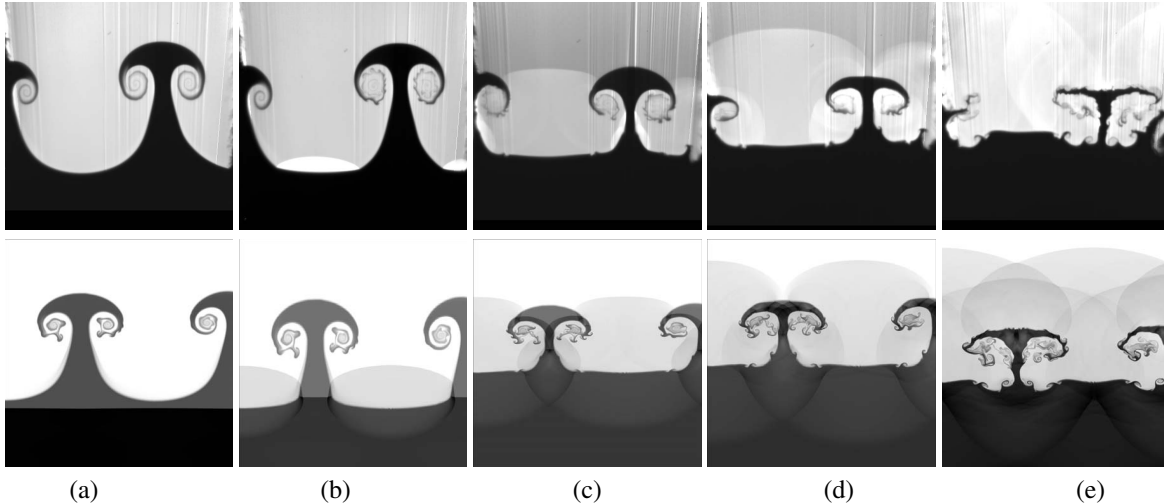
Fig. 1. Top row - the experimental images. Bottom row - the corresponding simulation images.

to collapse, leading to the mixing of the two fluids.

The experimental images are of varying sizes, with a typical size of 460×460. The simulation images are much larger. Though they are uniform in size for all time steps, the images shown here have been cropped to a size of 1330×1280 to focus on the region of interest. It is clear from these images that a comparison using a one-to-one correspondence is not possible. The sizes of the images are different and the structures of interest are in different locations. There is also noise in the experimental images in the form of vertical lines which may be lighter or darker than the surrounding regions. These noise streaks are typical of PLIF images. Further, there is no structure visible in the darker regions of the experimental images, while in there are clearly defined structures in the corresponding parts of the simulation. This is the result of the experimental setup.

We plan to compare the images over time by extracting higher level features from the data. Focusing on the single complete mushroom, we can characterize it using features such as the height of the mushroom, the width and height of the cap, and the width of the stem, as shown in Fig. (2).

There are several challenges to this feature extraction. First, in some of the experimental images, the noise is as strong as the signal (see for example the vertical lines in panel (c), top row, Figure 1). Thus, using an edge detector to separate the darker mushroom from the lighter background will also pick up these strong edges due to the noise. As this noise is quite structured, we cannot use any of the traditional denoising techniques which are more suitable for noise which is random in either pixel value or location such as Gaussian noise or salt-and-pepper noise. For example, if we use a 1-D median filter of a small size (say 1×5), it does not remove the vertical lines as many of them have a width greater than 5 pixels. If instead we use a wider filter (e.g. 1×10), then the noise is removed, but other details of the images are blurred.
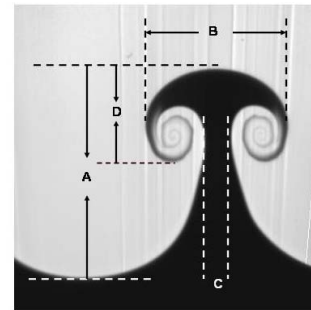
Fig. 2. The features used to represent the mushroom include (A): the height of the mushroom, (B): the width of the cap, (C): the width of the stem, and (D): the height of the cap.

Therefore, the first step is to identify a denoising technique specific to the characteristics of the noise in PLIF images.

3. DENOISING THE EXPERIMENTAL IMAGES

One problem domain with noise similar to the lines in PLIF images is the restoration of archival films. Line scratches, in the form of a vertical lines of bright or dark intensity oriented more or less vertically across the image, are a common form of degradation in such film. Automated denoising is done in two stages - an initial line detection step, followed by an interpolation or reconstruction step. Deterministic and stochastic approaches to detect these lines are described in [3] while the recontruction using Bayesian analysis is described in [4].

As our goal is not to restore the image, but to suppress the noise so that we can extract the mushroom structure from the background, we use a simpler version of this approach:

- **Step 1** - Enhance the image to improve the contrast using the following sharpening filter:
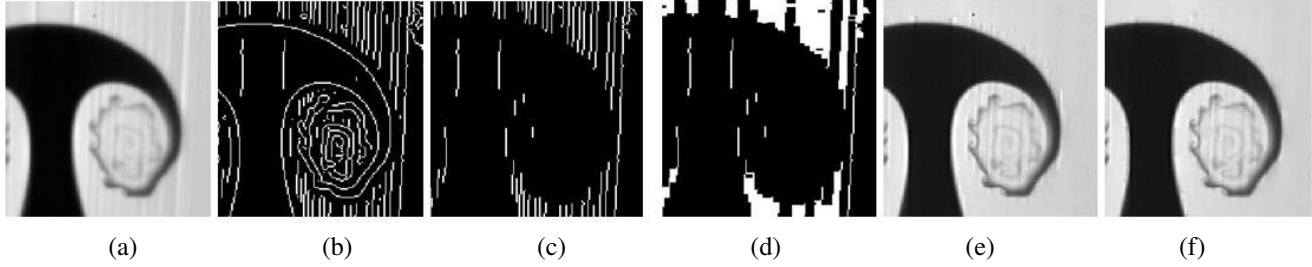
Fig. 3. Illustrating the process of denoising of the original image (a) after steps 2, 4, 5, 6 and 7 (panels (b) through (f)).

-0.6667	-1.6667	-0.6667
-1.6667	+4.3333	-0.6667
-0.6667	-1.6667	-0.6667

- **Step 2** - Use a Canny edge detector to produce a binary mask of all the edges.

- **Step 3** - Segment the binary image into multiple objects using a connected component algorithm with 8-connectivity of the edge pixels.

- **Step 4** - Identify the noise lines by keeping those objects whose bounding box ratio of width/height is less than 0.1. Since all these lines are almost vertical, we expect this ratio to be small for the lines, but not other objects in the edge image. The threshold value of 0.1 is chosen heuristically based on the experimental images.

- **Step 5** - If two lines are spatially close to each other, e.g. less than 4 pixels apart, then consider all the pixels between the two lines as noise pixels. As the noise lines are more than a single pixel wide, the Canny edge detector results in two thin lines representing the two edges of the wide line. This step merges these two thin lines back into one thick line, resulting in a mask image representing the noise lines in the image. Pixels outside the mask are labeled 0, those inside the boundary of the lines in the mask are labeled 1, and those on the boundary of the lines in the mask are labeled 2.

- **Step 6** - Create a denoised images as follows. First, apply a median filter of size 5×15 to the original image to smooth all the lines (and the mushroom boundaries as well). If the mask pixel has a value 0 or 2, set the corresponding pixel in the denoised image to the same value as the original image, else set it to the value in the median-filtered image. This step ensures that only pixels within the unwanted lines, but not on the boundary of the lines, are affected by the median filter.

- **Step 7** - At the pixels in the denoised image corresponding to mask pixels with value 2, apply an average filter of size 5×5. These are the pixels at the boundary of the line segments. This step effectively smooths the
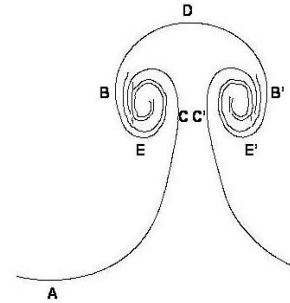
Fig. 4. The points on the edge image that can be used to extract the features for the mushroom structures).

discontinuities between the original image pixels and the the median-filtered image pixels.

4. FEATURE EXTRACTION

To extract the afore-mentioned features from the denoised experimental and the simulation images, we first obtain an edge image and identify the key points shown in Figure 4. This is done by borrowing an idea from optical character recognition where characters are recognized by considering their horizontal (vertical) projection profiles [5]. These are the histograms of the number of edge pixels along horizontal(vertical) scan lines. However, instead of considering the histograms themselves, we consider variations in the Number of Edge Pixels in a Column (Row), hereafter abbreviated as NEPC (NEPR).

For example, Point A can be calculated as the point where the NEPC is 1 and the y-location of the edge pixel is minimum. When we move from point A to the right in the image, point B is where the NEPC increases from 1 to a number greater than 1. Then, as we continue to move to the right, point C is the X location where the NEPC decreases from a number greater than 1 to 1. Next, point C' is 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. Next, point B' is where the NEPC decreases from a number greater than 1 to 1. Point D is de-

	B_x	B_x'	C_x	C_x'	D_y	A_y	E_y	E_y'
E1	233	408	306	339	384	115	264	265
E2	231	412	309	338	376	115	252	253
E3	233	409	309	325	330	147	221	215
E4	238	409	313	333	309	168	232	229

Table 1. X or Y locations of the interest points on the mushroom (Figure (4)) for the experimental images.

	HEIGHT	CAP HEIGHT	STEM WIDTH	L. CAP HEIGHT	R. CAP HEIGHT
E1	270	176	34	121	120
E2	262	182	30	124	124
E3	184	177	17	110	116
E4	142	172	21	78	81

Table 2. Mushroom features (in pixels) for the experimental images.

	B_x	B_x'	C_x	C_x'	D_y	A_y	E_y	E_y'
S1	182	664	381	470	1171	448	845	873
S2	179	667	381	470	938	328	609	633
S3	183	667	402	448	1090	636	829	841
S4	175	673	403	446	983	596	762	777

Table 3. X or Y locations of the interest points on the mushroom (Figure (4)) for the simulation images.

	HEIGHT	CAP HEIGHT	STEM WIDTH	L. CAP HEIGHT	R. CAP HEIGHT
S1	724	483	90	327	299
S2	610	489	90	330	305
S3	455	485	47	262	250
S4	388	499	43	222	206

Table 4. Mushroom features (in pixels) for the simulation images.

fined as the highest y-location of an edge pixel as we move to the right from B to B'. Points E and E' can be calculated in a manner similar to the calculation of B, except we use the NEPR instead of the NEPC.

A similar approach to calculating the mushroom features can be used for the simulation images. 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 NEPC 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.

The locations of interest points in the edge image are given in Tables 1 and 3 for experiments and simulations, respectively. The corresponding mushroom features are given in Tables 2 and 4. We do not consider the images in panel (e) of Figure 1 as the mushroom structure has collapsed by then.

Note that a direct comparison of the values in Tables (2) and (4) is not advisable as the pixels sizes are not the same between the experiments and the simulations. However, we can compare the structures by considering 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.

5. SUMMARY AND CONCLUSIONS

In this paper, we have described the use of image processing techniques to address an important problem in computer simulations, namely, validating the simulations by comparing them to experimental data. Through the use of denoising and feature extraction, we have shown that image analysis techniques enable a quantitative and objective comparison. They also hold the promise of automation, which is desirable as the simulation data can be quite large, making a manual analysis infeasible.

7. REFERENCES

[1] N. Brouillette, "The Richtmyer-Meshkov instability," *Ann. Rev. Fluid Mech.*, vol. 34, pp. 445–468, October 2002.

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

[3] A. C. Kokaram, *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*, Springer-Verlag, 1998.

[4] A. C. Kokaram, "Removal of line artefacts for digital dissemination of archived film and video," in *Proceedings of the IEEE Conference on Multimedia Computing and Systems, Volume 2*, June 1999, pp. 245–249.

[5] H. S. Hou, *Digital Document Processing*, John Wiley, 1983.