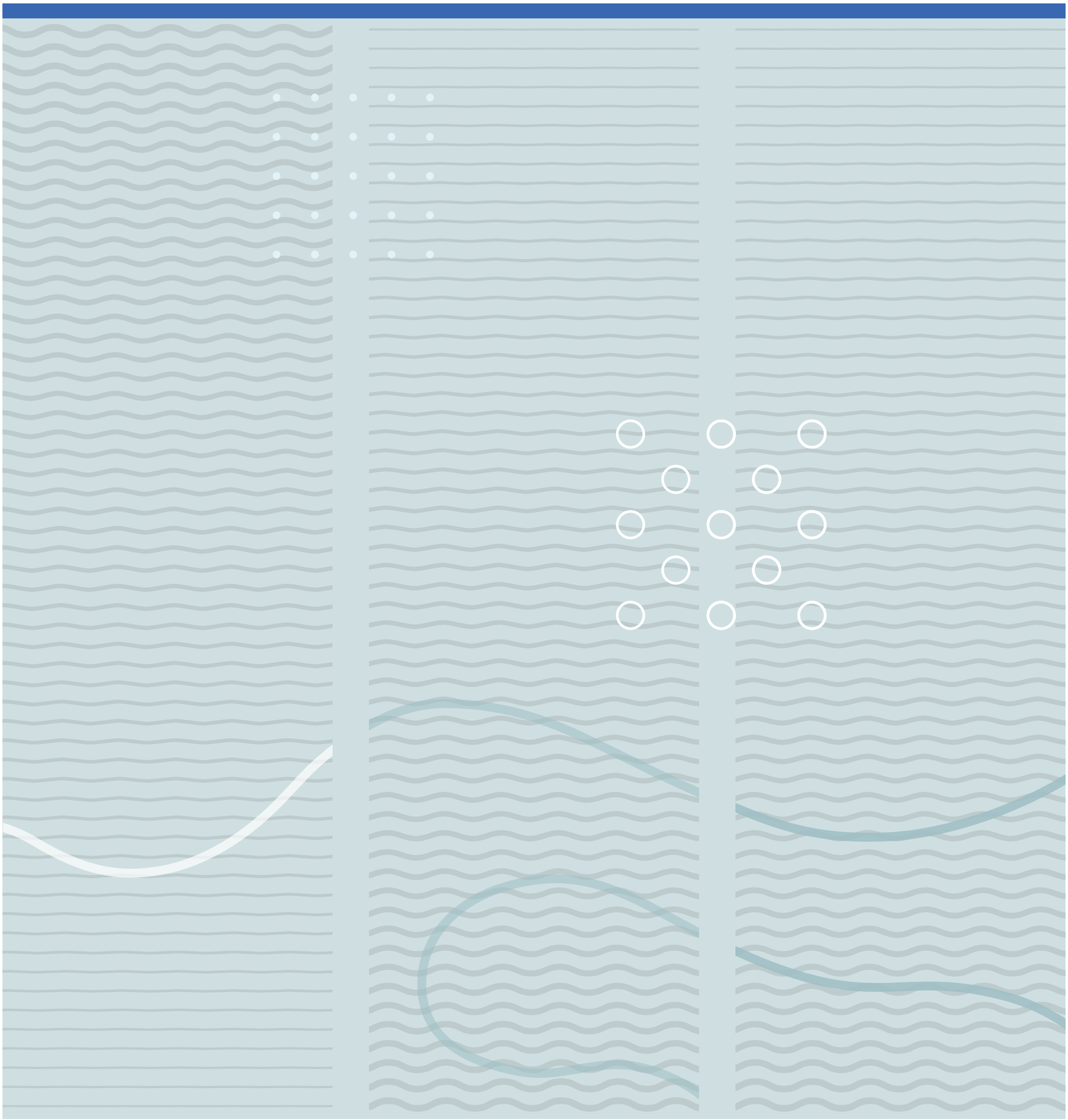
University of South-Eastern Norway
Faculty of Technology, Natural Sciences and Maritime Studies
—
Doctoral dissertation no. 83
2020

Samee Maharjan

An Image Processing Framework for High Speed Videos from Combustion and Gas Explosion Experiments



Samee Maharjan

An Image Processing Framework for High Speed Videos from Combustion and Gas Explosion Experiments

A PhD dissertation in

Process, Energy and Automation Engineering

Faculty of Technology, Natural Sciences and Maritime Studies
University of South-Eastern Norway
Porsgrunn, 2020

Doctoral dissertations at the University of South-Eastern Norway no .83

ISSN: 2535-5244 (print)
ISSN: 2535-5252 (online)

ISBN: 978-82-7206-579-8 (print)
ISBN: 978-82-7206-580-4 (online)

Print: University of South-Eastern Norway

To my beloved grandmother Lok Maya Maharjan..

Preface

This dissertation is submitted to the Faculty of Technology, Natural Sciences and Maritime Sciences, University of South-Eastern Norway (USN) in partial fulfillment of the requirements for the degree of Doctor of Philosophy. This thesis presents my four year PhD work about various image processing techniques to process the high speed videos from detonation and shock wave boundary layer interaction (SWBLI) experiments. Most of the experimental work included in this PhD were conducted at the California Institute of Technology (Caltech), California USA as a collaborative work between Caltech and USN.

I would like to express my sincere gratitude to my supervisors; Prof. Ola Marius Lysaker, Prof. Dag Bjerkevedt and Prof. Knut Vaagsaether. This research would not have been possible without clear guidance and continuous support from them. During the study, I got an opportunity to be part of a three month long program 'Geometry and Learning from Data in 3D and Beyond', at the Institute for Pure and Applied Mathematics (IPAM) at University of California, Los Angeles (UCLA). The program was supported by the National Science Foundation (Grant No. DMS-1440415).

My warm acknowledgment goes to the process safety group at USN: Andre, Joachim, Per Morten, Mathias for their valuable suggestions and fun time. I would also like to thank my fellow PhD students Fasil, Asanthi, Anirudh and friends Ananta, Prerana for making even the darkest of the days bright ones. I am extremely grateful to my dearest Zia for all the endless love and support throughout these years.

At last but not the least, I am forever indebted to my parents Shambhu Ram Maharjan and Meena Maharjan for allowing me to choose my way and helping to be where I am today. Thanks to my sister Lisha, brother Samin and all my extended family for never making me feel alone and giving me all the love.

Note: Article IV is now published in Signal, Image and Video Processing.

Porsgrunn, 25th October 2020

Samee Maharjan

Summary

The method of high speed photography is used to visualize those phenomena which occur so fast that it is impossible to visualize by the normal human eyes. One of those phenomena is a shock wave propagation during gas explosion experiments. A shockwave is a strong compression wave existing in the supersonic flow field across which gas properties like pressure, temperature, and density increase significantly. This thesis is aimed at developing image processing frameworks, which will process the high speed videos captured during gas explosion experiments and extract some useful information about the shock waves.

One way to extract any sort of information about propagating waves from the high speed videos is by tracking the position of the wave front. The common choice of image processing technique to perform this then naturally becomes any kind of edge detection technique. However, when the images are comparatively of low quality in terms of contrast, resolution and include a high amount of noise, basic edge detection techniques might not give the precised result. Hence, some of those image processing techniques, which have the potential to detect edges in the low quality images were studied and implemented in this thesis.

The first approach is based on a energy minimizing curve, which moves towards the edges and eventually lies around the edges, widely known as active contour models or Snakes. Based on a classic close contour approach, an open contour model is developed and implemented to contour a wave front from top to bottom of the image (Article I). The second technique studied is region wise image segmentation method called the watershed algorithm (Article II). Both of these methods do track the edges within a required precision however, they are time consuming. The active contour model requires a good initialization of the curve from the user and also includes multiple parameters. The watershed algorithm eliminates any parameter requirements however, requires multiple pre/post-processing steps.

The third technique is a statistical object detection method of template matching, which reduces the number of pre/post-processing steps. At first, a binary template matching is implemented in the binary images of the high speed videos (Article IV). This method minimizes the edge detection error but requires an image to be transformed into a binary form. The updated template matching uses a dynamic template that varies its intensity values depending on each considered image (Article III). This approach eliminated the need of image thresholding in order to detect edges. Furthermore, it shows to be more robust and faster compares to previous techniques. Even though it is possible to track the front without any pre-processing by using a dynamic template matching, it shows better results in the filtered image.

The images from a high speed camera, when operated in a higher frequency are comparatively of low quality, and also ongoing chemical changes in the flow significantly corrupt the images. A standalone edge detection technique therefore might not be able to track the front as accurately as when it is combined with an image denoising/filtering. Hence, image filtering in both spatial and Fourier domain were also studied and implemented before applying any of above mentioned tracking techniques.

The tracking of wave fronts do not only show the structure and position of the waves but also gives a possibility to extract primary information about the shock wave like, shock speed, shock angle, etc. Furthermore, secondary information like Mach number, pressure, and temperature can also be estimated, by combining the primary information and the traditional gas dynamics equations. While calculating shock speed from the tracked shock position, a basic two-point method (distance/time) shows some oscillations in the result. Hence, a relatively new approach non-linear square fit method (NLSFM) was modified and implemented, which reduces the oscillations significantly (Article V).

For validation, the estimated pressure for some of the experiments was compared to the reading from pressure transducers, which shows a good match. The results provide insightful information about the reflected shock wave and its boundary layer interactions. The calculated wave properties demonstrate a variation that occurred within a time interval of 300 microseconds (μs) at a distance of 100 millimeters (mm). This information is difficult to extract while using a traditional approach such as pressure transducers. Thus, a combination of the high speed videos and digital image processing has a huge potential to study gas dynamics phenomena is a detailed manner.

Contents

List of Figures

Publication

First author:

Article I: Open Active Contour Model For Front Tracking Of Detonation Waves. *Maharjan, S., Gaathaug, A.V., Lysaker, O.M.*: In: Proceedings of the 58th Conference on Simulation and Modelling, pp. 174–179. Linköping University Electronic Press, Sweden (2017)

Article II: An Image Processing Framework for Automatic Tracking of Wave Fronts and Estimation of Wave Front Velocity for a Gas Experiment. *Maharjan, S., Bjerketvedt, D., Lysaker, O.M.*: In: Representations, Analysis and Recognition of Shape and Motion from Imaging Data. RFMI 2017. Communications in Computer and Information Science, Springer-Verlag, (2019)

Article III: Wave Front Tracking in High Speed Videos Using a Dynamic Template Matching. *Maharjan, S.*: In: Pattern Recognition and Image Analysis. Springer-Verlag, pp. 531–542 (2019)

Article IV: Processing of High Speed Videos of Shock Wave Boundary Layer Interactions. *Maharjan, S., Bjerketvedt, D., Lysaker, O.M.*: Production Process: Signal, Image and Video Processing, (2020)

Article V: Information Extraction from High Speed Videos of Reflected Shock Wave Interaction With Boundary Layer.*Maharjan, S., Bjerketvedt, D., Lysaker, O.M.*: Well prepared and will submitt to Shock Waves.

Co-Author:

Velocity and pressure along detonation fronts - Image processing of experimental results. *Gaathaug, A.V., Maharjan, S., Lysaker, O.M., Vaagsaether, K., Bjerketvedt, D.*: In: Proc. of the Eighth International Seminar on Fire and Explosion Hazards (ISFEH8), pp. 133–149. (2016)

Wave Front Tracking using Template Matching and Segmented Regression. *Siljan, E., Maharjan, S., Lysaker, O.M.:* In: Proceedings of the 58th Conference on Simulation and

Modelling, pp. 326–331. Linkoping University Electronic Press, Sweden (2017)

Abbreviation

ANN	Artificial Neural Network
BOS	Background Oriented Schilieren
BP	Breaking Point
BP_o	Optimum Breaking Point
BSDS	Berkeley Segmentation Data Set
BV	Bounded Variation
CCD	Charge-Coupled Device
CFD	Computational Fluid Dynamics
CMOS	Complementary Metal-Oxide Semiconductor
CNN	Convolutional Neural Network
DA	Denoising Autoencoder
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
DFT	Discrete Fourier Transform
FCN	Fully Connected Network
FFT	Fast Fourier Transform
FPS	Frames Per Second
GALCIT	Graduate Aeronautical Laboratories California Institute of Technology
GDT	GALCIT Detonation Tube
GPU	Graphics Processing Units
GVF	Gradient Vector Feild
IFFT	Inverse Fast Fourier Transform
LoG	Log of Gaussian
LPF	Low Pass Filter
MSE	Mean Square Error
NLSFM	Non-Linear Square Fitting Method
PDE	Partial Differential Equation
PIV	Particle Image Velocimetry
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
ROF	Rudin-Osher-Fatemi
SAD	Sum of Absolute Difference
SDA	Stacked Denoising Autoencoder
SIV	Schlieren Image Velocimetry
SSE	Sum of Squared Error
SWBLI	Shock Wave Boundary Layer Interactions
TNRD	Trainable Non-Linear Reaction Diffusion
TP	Triple Point
TV	Total Variational

Part I

Overview

1 Introduction

This chapter of the thesis contains a brief background of images and high speed imaging along with an introduction of digital image and digital image processing. The chapter also includes the objectives and the challenges of this thesis.

1.1 Background

History of imaging goes back to early 19th century, when images were captured on the silver coated plates and took hours to be completed. Throughout the 19th century, photography techniques kept on improving in minimizing, the time required, size of the apparatus and the labor needed. One of the major development was introduced in the year 1878 by Eadweard Muybridge. He introduced high speed photography, which captures those phenomena which occur so fast that it is impossible to visualize by the normal human eyes. He used high speed photography to determine whether a horse lifted all four hooves off the ground or not when galloping (Shimamura, 2015). A snippet from the first high speed film is presented in Figure 1.1.

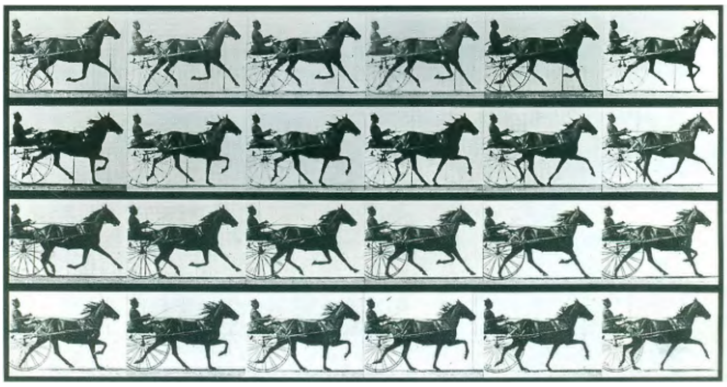


Figure 1.1: A series of images from the first high speed video captured by Eadweard Muybridge in 1878 (Shimamura, 2015).

After few years, physicist Peter Salcher captured the first image of a supersonic (speed greater than the speed of sound) bullet using high speed photography, which was later used by Ernst Mach to study the supersonic motion (Pohl, 2002). One of the early images of a flying bullet from the study of Ernst Mach is presented in Figure 1.2. In the field of gas dynamics/combustible flow, images have been part of flow visualization from the early years. They provided the needed visual representation of the phenomena for the researchers. Mainly two types of flow visualization techniques i.e. shadowgraph and schlieren were used for capturing the flows with variable density (Settle, 2001). However, the images were restricted mainly for visual purpose and any information from the images had to be extracted manually.

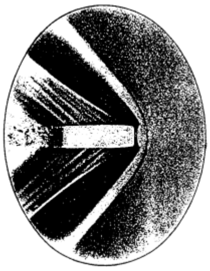
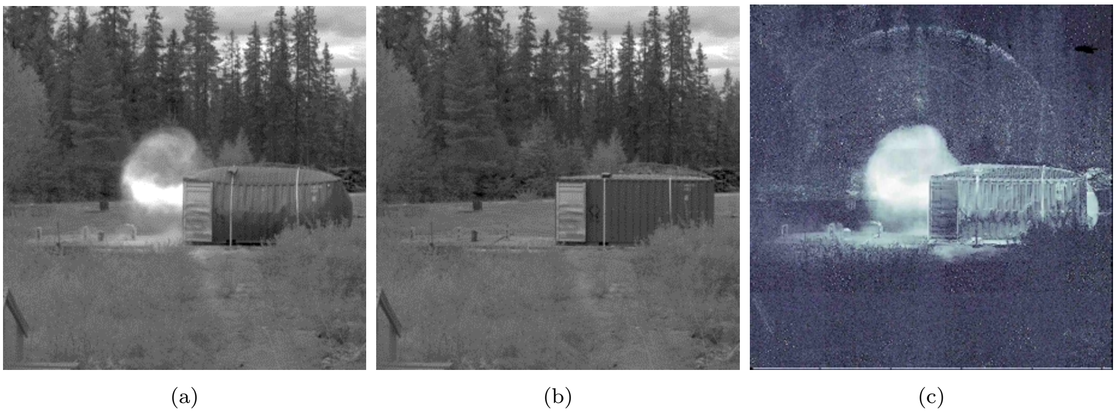


Figure 1.2: The first image capturing a flying bullet (Pohl, 2002).

A breakthrough in the imaging technology came by the introduction of a digital imaging around the late 1960s (Rosenfeld, 1969). Later during the 1980s and 1990s, the development of digital cameras along with charge-coupled device (CCD) and complementary metal-oxide semiconductor (CMOS) technology revolutionized digital photography and eventually high speed imaging. Today, there are high speed cameras that operate at more than a million frames per second (FPS). Following this development along with the easy availability of computers, investigations on extracting the information from the schlieren and shadowgraph high speed images using digital image processing techniques have been increasing (Kleine, 2005). However, in most of the work, image processing was not the main focus but used as a tool to validate experimental results (Sommersel et al., 2008; Damazo, 2013; Mata et al., 2017).

For example in Sommersel et al. (2008), a high speed imaging with a flow visualization technique called Background Oriented Schlieren (BOS) (Raffel, 2015) was used for finding the shock wave position in the images. The pressure of the wave was estimated by using the shock positions, which was later used to validate the experimental pressure records. An example of a background subtraction technique is presented in Figure 1.3. It can be observed from this example that, an image processing can extract the information which otherwise is not visible for human eyes. Thus, a combination of the high speed videos and digital image processing has a huge potential to study gas dynamics apart from traditional experimental studies and computer simulations.



(a) (b) (c)

Figure 1.3: Background Oriented Schlieren (a) distorted image; (b) undistorted image; (c) result of subtracting image (a) from (b), position of the shock wave becomes visible (Sommersel et al. (2008)).

1.2 Digital Image

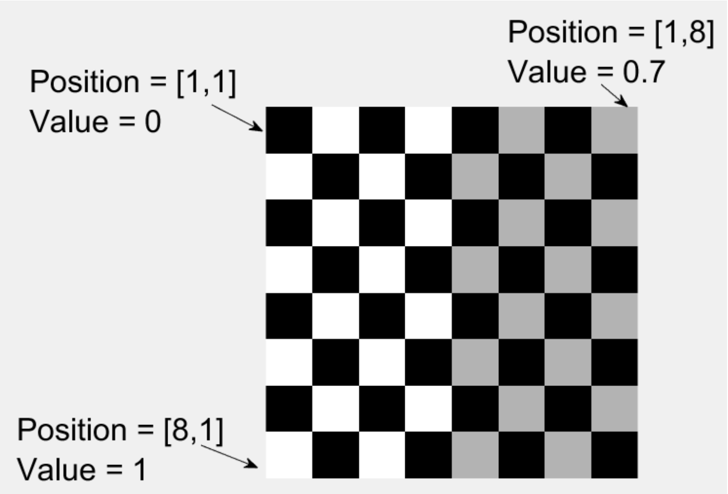


Figure 1.4: An example of a digital image showing the location and intensity of few pixels.

A two dimensional (2-D) image can be defined as a 2-D signal $f(x,y)$, where x and y are the spatial coordinates, and the value of $f(x,y)$ at any pair of coordinates (x,y) is the intensity of an image at those coordinates. If (x,y) and $f(x,y)$ are both finite and discrete quantities, we call it a digital image. Hence, a 2-D digital image is a matrix, composed of a finite number of elements called pixels, each of which has a particular position and intensity. An example of a 2-D digital image whose pixel values are normalized between 0-1 is shown in Figure 1.4. As can be observed from Figure 1.4, image/pixel coordinate system is slightly different from normal (x,y) coordinate system. It defines x-axis as a column, increasing from left to right and y-axis as row increasing from top to bottom keeping the origin at the top left corner. A pixel position is therefore generally defined by its row and column position, for example, the bottom left pixel position in Figure 1.4 shows the location of row number 8 and column number 1. A digital image can be of various types depending on the pixel values, however this thesis mainly deals with 1) grayscale image which pixel value lies between 0 - 255 and 2) binary image whose pixel value is either 0 or 1.

Digital images can be read and processed using computer algorithms, which is basically known as a digital image processing. It was introduced in the late 1960s, however, the process was expensive and the application was limited to few fields like satellite imagery (NASA, 1999). The introduction of fast and cheap commercial computers around the mid 1990s expanded the area of research using computer algorithms. At present, the application is expanded widely in the field of computer vision (Arnold et al., 2019), medical imaging (Hoheisel, 2006) etc.

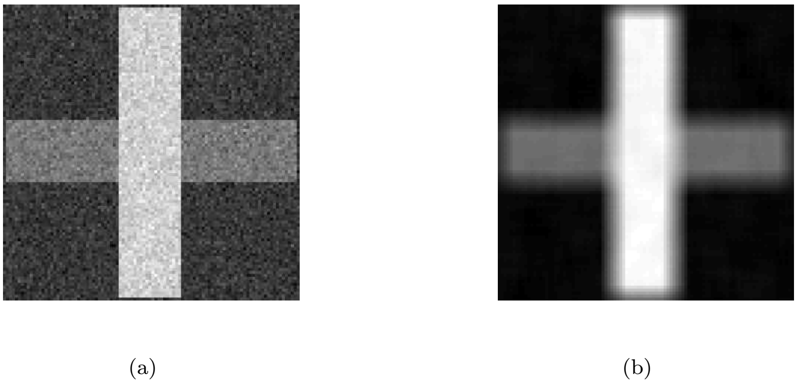
Onwards in this thesis, digital image and digital image processing referred as image and image processing respectively.

1.3 Image processing techniques

There are numerous image processing techniques, for example, image filtering/denoising, image enhancement, image segmentation, image classification and edge detection to mention few (Gonzalez and Woods, 2000; Chan and Shen, 2005). Most of the image processing techniques are implemented based on methods like;

- Statistical: based on probability and statistics.
- PDE: based on energy minimization and iterations.
- Domian transform: images are transformed into other domains apart from the spatial domain for the processing.
- Machine learning: use a machine to make a decision based on training.

The short introduction of two most important image processing techniques for this work i.e. image denoising and edge detection are described in the following subsections.



(a)

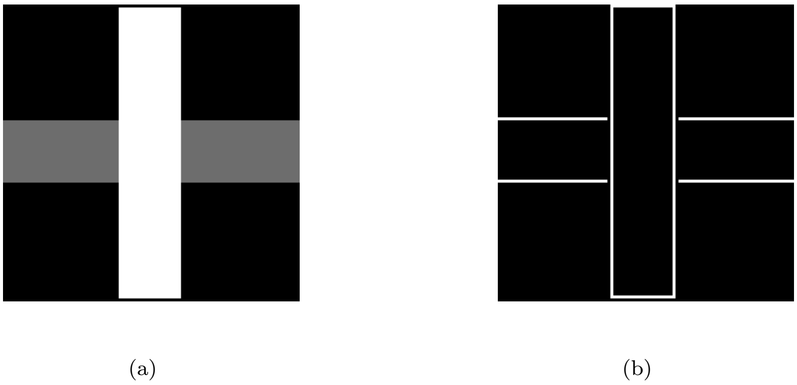
(b)

Figure 1.5: Image denoising (a) grayscale image degraded by random noise; (b) result after image denoising.

1.3.1 Image denoising/filtering

Image denoising is a process of eliminating random noise as far as possible while preserving important details of an image. A noisy image produces undesirable visual quality, besides it also lowers the visibility of low contrast objects. Hence, noise removal is an essential part to recover and enhance fine details that are hidden in the image data (Gonzalez and Woods, 2000). Image filtering is one of the common ways to reduce or eliminate noise from the images. The process can be defined as an operation in which the value of an output pixel is determined by a combination of the pixel values in the neighborhood of the corresponding input pixel, which results in a smooth image. An example of image filtering is depicted in Figure 1.5.

1.3.2 Edge detection



(a)

(b)

Figure 1.6: Edge detection (a) grayscale image; (b) result of edge detection.

Edge detection is one of the fundamental image processing techniques, for finding the boundaries of objects within an image. It is implemented by identifying sharp discontinuities in the image intensities as can be seen in Figure 1.6. It plays a significant role in image processing applications like image segmentation (Senthilkumaran and Reghunadhan, 2007) and motion tracking (Murray and Basu, 1994). Motion of any object in a series of images can be tracked by detecting the edges of that object in those images. This is one of the reason why edge detection is an intergral part of this thesis. Moreover, edge detection usually results as a binary image, which reduces a memory size of the image significantly while preserving important features.

1.4 Objective and Problem Description

High speed videos were recorded during two types of gas experiments, i.e, detonation and SWBLI experiments. Gas detonation experiments were conducted in the setup located at USN and, SWBLI experiments were conducted at GLACIT detonation tube (GDT), Caltech. The experiments were recorded in the high speed videos by using a high speed camera named KIRANA[1]. The camera can take upto 5 millions FPS, however it was operated with the frame rate of either 500,000 or 200,000 FPS. The flow visualization techniques, shadowgraph or schlieren were used for capturing the flow (Settle, 2001).

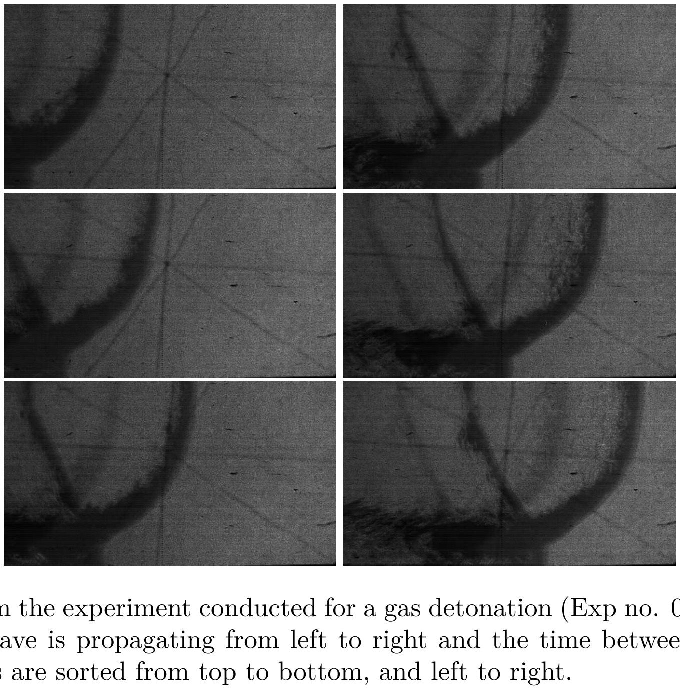


Figure 1.7: Images from the experiment conducted for a gas detonation (Exp no. 00022) in 30% H_2 above air. The wave is propagating from left to right and the time between each image is 24 μs. The images are sorted from top to bottom, and left to right.

[1] https://www.specialised-imaging.com/products/video-cameras/kirana

An example of a gas detonation experiment, showing the detonating H_2 gas above a non-reacting layer of air is presented in Figure 1.7. Generation of Mach stem around the lower boundary of the tube can be observed in the Figure 1.7. An exmaple from the SWBLI experiments conducted with CO_2 gas, showing a reflected shock wave is shown in Figure 1.8. Reflected shock wave initiates once an incident shock wave hits the end wall (Mark, 1958). Distortion of the reflected shock wave foot due to its interaction with a boundary layer created behind the incident shock can be seen in the Figure 1.8.

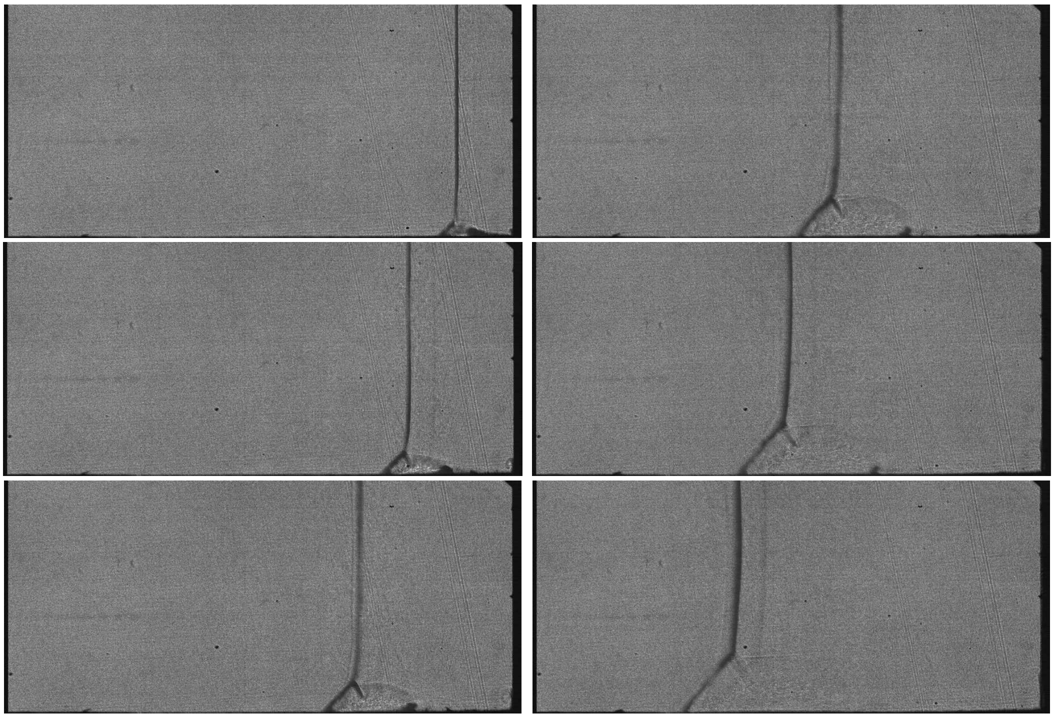


Figure 1.8: Images from the experiment conducted with CO_2 for SWBLI (Exp no. 2516). The wave is propagating from right to left and the time between each image is 40 μs. The images are sorted from top to bottom, and left to right.

The main objective of this PhD is to study image processing techniques for analysing these high speed videos and extract primary information like the shock speed, the position of triple point and the shock angle (refer Figure 1.9(a)). Thereafter, by combining the calculated primary information and traditional gas dynamic equations, secondary information like pressure and temperature can be estimated as well. At the end, the calculated results can be used for validating results from pressure tranducers and computer simulations.

One way of extracting primary information is by finding the position of a wave front in each image which is basically a front tracking, for instance like the yellow curve in Figure 1.9(a). For human eyes, it is a rather simple task to find the position of wave front in these images, however, a close look around the wave front in Figure 1.9(b), makes the

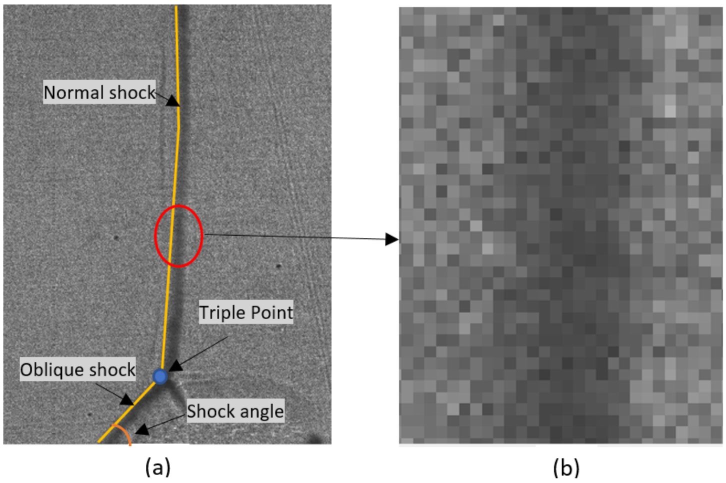
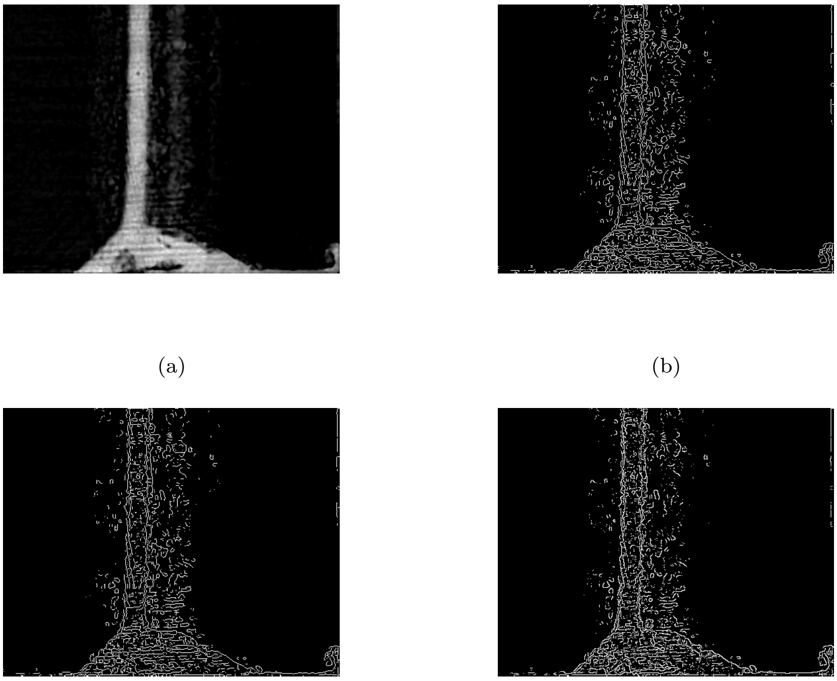


Figure 1.9: (a) An image from high speed video showing different parts of a reflected shock wave; (b) closed look at the red marked area around the wave front.

task subjective. In addition, if the number of images is large, manual tracking becomes inefficient. Therefore, image processing algorithms that estimate the wave front position as accurately as possible are required. And to find the position of any object in an image, the edge detection technique introduced in Section 1.3.2 naturally becomes a choice. Image filtering technique described in Section 1.3.1 is then used to help edge detection by removing any kind of background noise.

The images processed during this thesis are comparatively of lower quality in terms of object contrast, resolution, etc. This is mainly due to higher camera frequency, varying background noise, lighting. Hence, straight forward edge detecting techniques didn't perform as accurately as when applied to better quality images. For example, the result of applying one step edge detection techniques as in Guoshuai et al. (2019), are shown in Figure 1.10. As can be seen from the results, edge detections are not good enough to detect the position of the wave front. Hence, there is a necessity of developing front detection algorithms that could detect the wave front even in lower quality images.

Besides, some other challenges need to be addressed as well. Such as, the developed detection algorithms should be able to detect wave fronts in different videos with minimum to no change on any parameters. The wave structure is different in every high speed video, thus the algorithms should be independent of structure. At last, as this PhD is application driven, the algorithms should be easily understandable and applicable without being expert in image processing. There are numerous image processing methods purposed for edge detection in a noisy environment, still, they are seldom used in wave front detection. One of the reasons might be, the purposed image processing algorithms are relatively complex and require some expertise. With this thesis, some of the existing image processing techniques of edge detection, which have a potential on the wave front tracking in low quality or noisy images are explored.



(a)

(b)

(c)

(d)

Figure 1.10: Result of MATLAB edge detection function. (a) Filtered image; (b) Sobel; (c) Prewitt; (d) Roberts.

Once the wave front position is tracked, primary information like the shock speed (both normal and oblique), the position of triple point, and the shock angle (refer Figure 1.9(a)) need to be calculated. The challenge while estimating shock speed is the high sensitivity, meaning mispositioning of even one pixel will influence the speed calculation drastically. Usually, to overcome this challenge, the speed is estimated over a longer time, however, this will miss out on the actual variation happening in between. So, the methods which could overcome this challenge also need to be studied and implemented.

1.5 Thesis structure

This thesis is divided into three parts.

Part I is an overview of the PhD work and consists of multiple chapters, an introduction of the thesis including background and description of imaging, problem description, and the objective of the thesis is described in Chapter 1. The literature study in Chapter 2 describes the existing image processing methods for image denoising/filtering, edge detection, and object tracking. The details of experimental setups and high speed videos used for the thesis are explained in Chapter 3. Chapter 4 briefly explained the description of all the methods developed throughout the study. In Chapter 5, the results of methods are discussed according to the corresponding articles. For the details about the developed algorithms and their implementation, it is requested to refer to the corresponding articles. The conclusions are given in Chapter 6.

Part II consists of all the published and prepared scientific articles. In the end, Part III of this thesis contains the unfinished work of using Machine learning to track the wave front. Though, this work couldn't progress forward and isn't mature enough to draw any conclusion, it can be one of the interesting work in the near future.

2 Literature review

This chapter is divided into four sections; Section 2.1 and Section 2.2 is a selection of a literature study on various image denoising/filtering and edge detection techniques respectively. These sections are aimed for readers who are relatively new to image processing, thus the methods containinig from the basic one step to some advanced ones are included. These techniques are classified based on the different methods mentioned in Section 1.3. Section 2.3 gives a general overview of the progress of image processing techniques in the shock wave analysis. While the wave tracking in high speed images is not studied as it should be and there is not many work in literature, object tracking itself is a very well researched topic. There are numerous image processing methods purposed for different type of object tracking in a noisy environment, which implements relatively more advanced edge detection techniques. Some of those methods which are in the scope of this thesis are described in Section 2.4.

2.1 Image denoising/filtering

Classic image filtering techniques use a basic way of sliding a pre-defined kernel over an image, which smooths the overall image. Modern filtering techniques involve relatively advanced methods like PDEs and space transform, which smooth the noise while preserving the edges and other features. In this section, some of the techniques from both classic and modern methods are presented.

2.1.1 Statistical based techniques

Most of the statistical based techniques are performed by sliding the kernel (h) over the image, generally starting at the top left corner. It moves the kernel through all the positions where the kernel fits entirely within the boundaries of the image. Each kernel position corresponds to a single output pixel, the value of which is depends on which kind of filtering technique has been used.

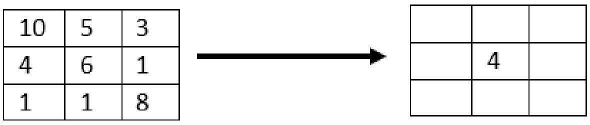
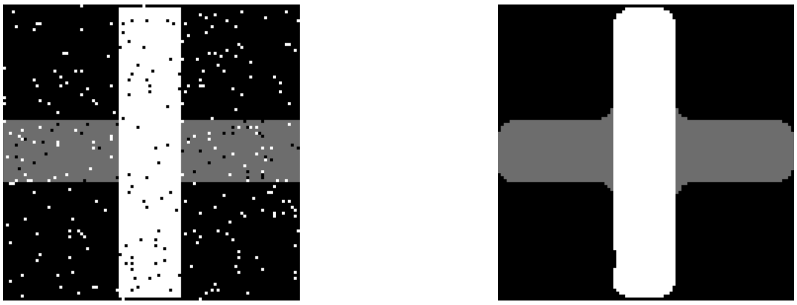


Figure 2.1: An operation of median filter of kernel size [3 × 3].

A **non-linear filtering** is one of the techniques often used to remove noise in an image. One of the popular non-linear filter is a median filter, whose output value is the middle element of a sorted array of pixel intensities from the pre-defined kernel as in Figure 2.1. Since the median value is robust to outliers, the filter is used for reducing the impulse noise (Yang et al., 1995). The result of using median filter of kernel size $[9 \times 9]$ onto a grayscale image of size $[100 \times 100]$ degraded by salt and pepper noise is presented in Figure 2.2. Some of the other non-linear filters are Bilateral filters, Anisotropic diffusion, Morphological operations (on binary images) (Lim, 1990), Rank conditioned rank selection (Hardie and Barner, 1994) etc.



(a)

(b)

Figure 2.2: Median filter (a) grayscale image degraded by salt and pepper noise; (b) filtered image with [9 × 9] kernel size median filter.

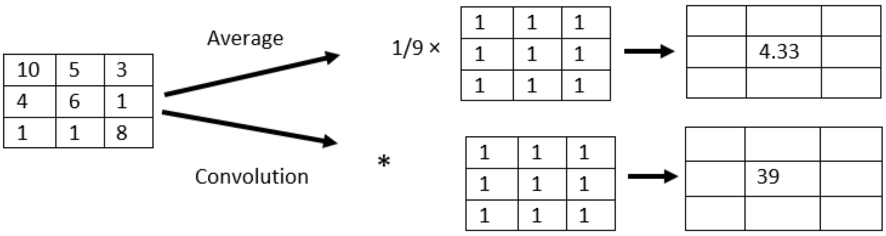
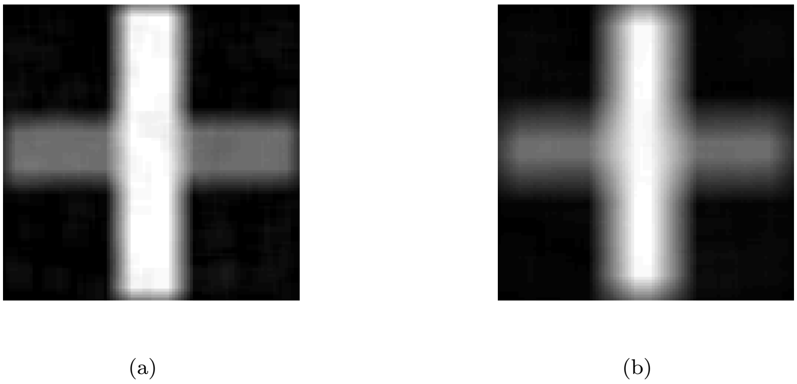


Figure 2.3: An operation of [3 × 3] linear filter; average filter operation and convolution operation.

Linear filtering is a filtering algorithm in which a pixel value in an output image is a linear combination of the neighbouring pixel values in an input image. The simplest linear filter is a mean/average filter, whose output is an average of the intensities of the pixels that surround the considered pixel. In practice, however, linear filtering is achieved rather by a convolution (Lim, 1990). The output pixel value is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together as in (2.1), a is the size of kernel here. An example of an average filter and a convolution operation is illustrated in Figure 2.3. The result of using convolution on Figure 2.2(a) with two different kernel size is presented in Figure 2.4.

$$I_f(m,n) = \sum_{i=-a}^{a} \sum_{j=-a}^{a} h(i,j) I(m-i, n-j). \tag{2.1}$$


(a)

(b)

Figure 2.4: Result of convolution corresponding to Figure 2.2(a) with kernel size of; (a) [9 × 9]; (b) [16 × 16].

Apart from above described techniques, statistical based techniques like Bayes theorem and maximum likelihood are also used massively in the field of image processing. The application might not be particularly for filtering, however these methods are huge part of image analysis specially image segmentation and reconstruction.

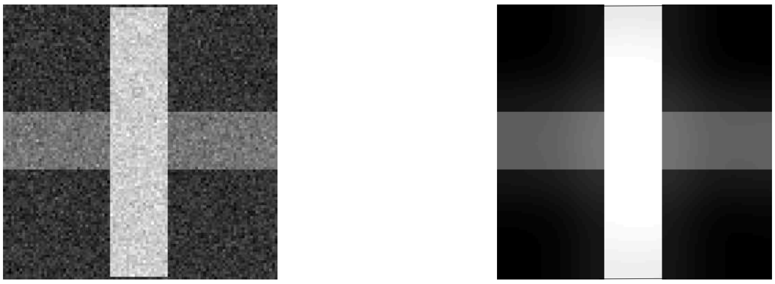
2.1.2 PDE based techniques

Total variation (TV) denoising is one of the prominent PDE based denoising techniques. It is established on the observation that the noise will be at high frequencies and the images with excessive detail will have a high total variation i.e. the integral of the absolute gradient of those images is high. Based on these observations, it is proposed to reduce the total variation of the image subject to getting a close match to the original image (Rudin et al., 1992) see also Chan et al. (2006) for a review of advances and literature.

It claims to be better than a conventional linear and median filtering, as it prevents the edges while smoothing noise. Suppose that we are given a noisy image I, then denoised image I_f can be obtained as the solution of a minimization problem,

$$\underset{I_f \in BV(\Omega)}{\operatorname{argmin}} \parallel I_f \parallel_{TV(\Omega)} + \frac{\lambda}{2} \int_{\Omega} (I - I_f)^2 dx, \tag{2.2}$$

where λ is a positive parameter, $BV(\Omega)$ is the bounded variation over the domain Ω, $TV(\Omega)$is the total variation over the domain and $\parallel . \parallel$ is the Euclidean norm. This minimization problem is referred to as the Rudin-Osher-Fatemi or ROF problem.



(a)

(b)

Figure 2.5: TV denoising (a) grayscale image with a random noise; (b) result after TV denoising.

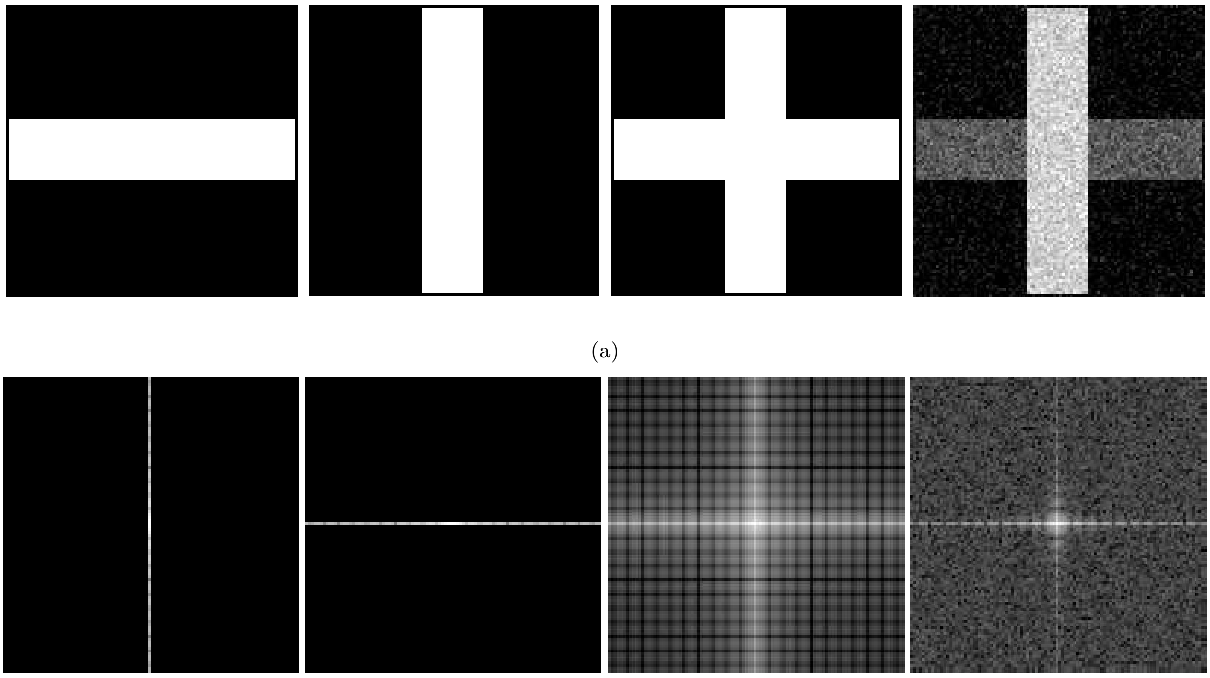
Split Bregman method is a technique for solving a variety of L1-regularized optimization problems and is particularly effective for problems involving TV regularization (Tom and Osher, 2009). The results using this method for TV denoising are presented in Figure 2.5. Another popular PDE based method for smoothing images is a **Linear diffusion** methods, which assume the intensity of illumination on edges varies like geometric heat flow in which heat transforms from a warm environment to a cooler one until the temperature of the two environments reaches a balanced point (Perona and Malik, 1990). Several authors have proposed higher order PDEs for image denoising and edge detection (You and Kaveh, 2000; Lysaker et al., 2003).

2.1.3 Transform based techniques

Image filtering using a convolution of a kernel in each pixel becomes operationally costly when the image size is too large. Therefore, in such cases, filtering can be done in a frequency/Fourier domain, where the input image can be directly multiplied with a filter function. **Fourier transform (FT)** is a mathematical technique for transforming a spatial domain signal into a frequency domain signal. The main concept is, in frequency domain low frequency correspond to a smooth varying image features whereas, high frequency corresponds sharply changing features such as boundaries or edges. A 2-D discrete Fourier transform can be obtained as,

$$F(k,l) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}. \tag{2.3}$$

Here, F and I is a Fourier transformed and spatial image respectively with size $[M,N]$ and $i = \sqrt{-1}$.



(a)

(b)

Figure 2.6: FFT of images (a) grayscale images in spatial domain; (b) corresponding Fourier transformed images.

Some examples of spatial images and the corresponding Fourier transformed images are shown in Figure 2.6. Once the image is transformed into Fourier domain, it is multiplied with the filter function H as in (2.4) to get the filtered image F_f. Thereafter, F_f is transformed back to the spatial domain filtered image I_f by using inverse Fourier transform, as in (2.5),

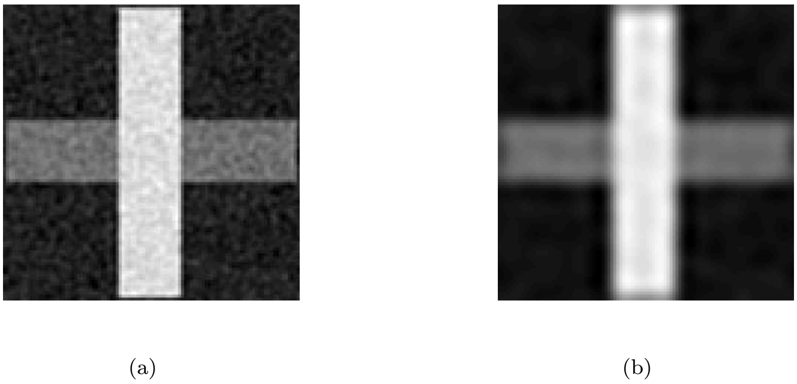
$$F_f(k,l) = F(k,l) \times H(k,l). \tag{2.4}$$

$$I_f(m,n) = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} F_f(k,l)\, e^{j2\pi\left(\frac{k}{M}m+\frac{l}{N}n\right)}. \tag{2.5}$$

A common way to filter the noise from an image in the frequency domain is by using low pass filter (LPF), which attenuates high frequencies greater than a given cut off frequency f_c, resulting in a smoother image in the spatial domain. The LPF function is given by,

$$H(k,l) = 1 \quad \text{if } \sqrt{k^2+l^2} < f_c, \tag{2.6}$$

$$H(k,l) = 0 \quad \text{if } \sqrt{k^2+l^2} > f_c. \tag{2.7}$$



(a)

(b)

Figure 2.7: Frequency filtering of the top right grayscale image in Figure 2.6 with low pass filter of cut off frequency (a) 30 Hz; (b) 10 Hz.

Filtered images by using fast Fourier transform with a cut off frequency of 30Hz and 10Hz can be seen in Figure 2.7. Further, on the fast Fourier transform and filtering of images in the frequency domain can be read in Najim (2010). The advantages of using Fourier analysis to study the shock wave unsteadiness can be read in Estruch et al. (2008).

Another image filtering technique which operates in a different domain apart from the spatial domain is **wavelet transform**. This transform decomposes a signal with finite energy in the spatial domain into a set of functions as a standard in the modular spatial domain of orthogonal. Compared with the traditional Fourier analysis, the wavelet transform can analyse the function in the modular spatial domain and timing domain which has a better local capacity of the frequency and time (Li, 2003; Xizhi, 2008). A two stage non-locally collaborative filtering method in the transform domain is BM3D purposed by Dabov et al. (2007). In this method, similar patches are stacked into 3-D groups by block matching, and the 3-D groups are transformed into the wavelet domain.

2.1.4 Machine learning

With the continuous development of machine learning, more and more image processing problems were being solved by various types of networks and achieved good results. For image denoising also, numerous approaches have been purposed in the past few years. In Jain and Sebastian (2009), a convolutional neural network (CNN) model was used for denoising images. Their algorithm claims to achieve better results than the traditional models. Denoising based on stacking layers of denoising autoencoders which are trained locally to denoise corrupted versions of their inputs was purposed in Vincent et al. (2010). A different kind of neural network with a special architecture (i.e. containing a sparsifying logistic) and image patches is used in Ranzato et al. (2007) to the denoise image. Chen and Pock (2017) proposed a feed-forward deep network called the trainable non-linear reaction diffusion (TNRD) model, which achieved a better denoising effect. Zhang et al. (2017) introduced residual learning and batch standardization into image denoising for the first time; they also proposed feed-forward denoising CNNs (DnCNNs).

2.2 Edge detection

There are various ways of detecting edges, one simple way is by approximating the first or the second order derivative of image intensity. A relatively advanced way is by using an energy minimizing curve, which eventually lies around the edges. Similarly, finding regions in an image whose pixels have the same properties instead of detecting sharp changes, is a region based way of detecting edges. The region based methods are extensively used for image segmentation.

2.2.1 Gradient based techniques

One of the usual approaches to find the magnitude and the direction of the intensity changes in an image I is a gradient operator (Gonzalez and Woods, 2000), defined as the vector,

$$\nabla I = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \tag{2.8}$$

The magnitude (M) and non-unique direction (α) of the gradient vector ∇I are calculated as,

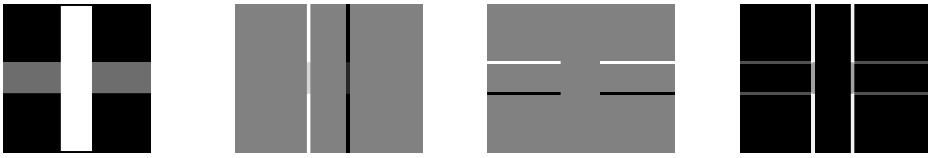
$$M(x,y) = \sqrt{g_x^2 + g_y^2}, \tag{2.9}$$

$$\alpha(x,y) = tan^{-1}\left(\frac{g_x}{g_y}\right). \tag{2.10}$$

The simplest of all edge detectors that use first order derivative to identify the intensity change is **Roberts** edge operator, which was introduced in Roberts (1963). At any pixel position (m,n), it can be computed as,

$$\frac{\partial I}{\partial y} \approx I(m,n+1) - I(m,n), \tag{2.11}$$

$$\frac{\partial I}{\partial x} \approx I(m+1,n) - I(m,n). \tag{2.12}$$



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

Figure 2.8: Gradient of an image (a) grayscale image; (b) horizontal derivative g_x; (c) vertical derivative g_y; (d) gradient magnitude image M.

An example of the gradient (horizontally and vertically) and its magnitude by using the Roberts operator is presented in Figure 2.8. The Roberts operator can be implemented as a 1-D mask for vertical and horizontal edges as shown in Figure 2.9(a). An update or more symmetric version of the Roberts operator is **Prewitt** operator, which consider both sides of a central point more like a central difference (Prewitt, 1979). This operation can be implemented by filtering the image with two $[3 \times 3]$ masks as in Figure 2.9(b). A slight variation of the Prewitt operators which gives more weight on the central coefficients of the masks as in Figure 2.9(c) is **Sobel** operator (Sobel, 2014).

-1
1

-1	1

(a)

$-\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$
0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

$-\frac{1}{3}$	0	$\frac{1}{3}$
$-\frac{1}{3}$	0	$\frac{1}{3}$
$-\frac{1}{3}$	0	$\frac{1}{3}$

(b)

$-\frac{1}{4}$	$-\frac{1}{2}$	$-\frac{1}{4}$
0	0	0
$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

$-\frac{1}{4}$	0	$\frac{1}{4}$
$-\frac{1}{2}$	0	$\frac{1}{2}$
$-\frac{1}{4}$	0	$\frac{1}{4}$

(c)

Figure 2.9: (a) Robert operator; (b) Prewitt 2-D masks of size $[3 \times 3]$; (c) Sobel 2-D masks of size $[3 \times 3]$.

Apart from the first order, some operators are based on the second order derivative. These operators are based on detecting edges in images by finding zero crossing points of the second derivative/Laplacian of the image. To do this, the Laplacian of Gaussian (LoG) combines Gaussian filtering with the Laplacian (Marr and Hildreth, 1980; Fisher et al., 2003). The same filtered image $LoG(x,y)$ can be obtained by either first convolving the image with a Gaussian kernel and then approximating the Laplacian as in (2.13), or convolving the image with the linear filter that is the Laplacian of the Gaussian filter as in (2.14)

$$LoG(x,y) = \nabla^2[G(x,y) * I(x,y)]. \tag{2.13}$$

$$LoG(x,y) = [\nabla^2 G(x,y)] * I(x,y). \tag{2.14}$$

2.2.2 Transform based techniques

The classic **Hough transform** was first introduced to identify the lines in the image, but later it was extended to identifying positions of arbitrary shapes, most commonly circles or ellipses (Hough, 1962). In general, the straight line $y = ax + b$ can be represented as a point (b, a) in the parameter space. However, vertical lines pose a problem. They would give rise to unbounded values of the slope parameter a. Thus, for computational reasons, Duda and Hart proposed the use of the Hesse normal form,

$$r = x\cos\theta + y\sin\theta, \tag{2.15}$$

where r is the distance from the origin to the closest point on the straight line, and θ is the angle between the x-axis and the line connecting the origin with that closest point. It is, therefore, possible to associate with each line of the image a pair (r, θ). A line is detected if the number of points in a pair (r, θ) is above a suitable threshold. The (r, θ) plane is sometimes referred to as Hough space for the set of straight lines in two dimensions (Duda and Hart, 1972).

Fourier transform is also used in some cases to detect the edges in an image, which are usually made of high frequencies. In the case of FFT, a high pass filter keeps all high-frequency features (e.g. sharp peaks and corners) which are usually not classified as edges.

2.2.3 PDE based techniques

These techniques use one or more parametric curve which moves and changes shape and size according to some kind of energy defined by image intensity. They mostly rely on the strong edges as a stopping factor, thus contouring the edges.

Active contour is one of the widely researched PDE based methods, which was introduced in Kass et al. (1988) and commonly known as a snake model. The idea is to evolve a parametric curve initialized manually by a set of (x, y) points around the object of interest.

By representing the snake $V(s) = (x(s), y(s))$ in (x, y) coordinate system parameterized by arclength ($s \in [0,\ 1]$), its energy function is defined as in (2.16). The energy is calculated for each point along with the snake and the snake will move in the direction where there is minimum energy compare to the previous position. The snake will be stationary once the energy is balanced in all directions.

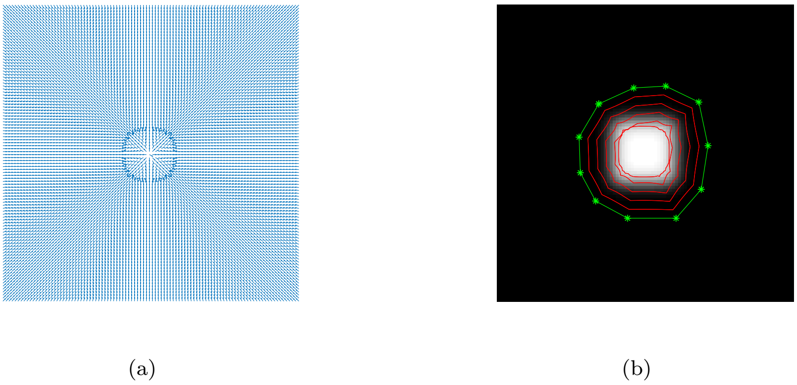
$$E_{snake} = \int_{s} E_{int}(V(s)) + E_{ext}(V(s))ds. \tag{2.16}$$

The first term in (2.16), E_{int} is an internal energy which is responsible for the smoothness of the curve, and can be defined as

$$E_{int} = 1/2[\alpha|V'(s)|^2 + \beta|V''(s)|^2]. \tag{2.17}$$

Here α and β are positive weighting parameters for controlling the snake's tension and rigidity respectively. $V'(s)$ and $V''(s)$ are the first and the second derivative of $V(s)$ with respect to s. The second term in (2.16), E_{ext} is the external energy which is responsible for attracting the curve towards the edges. It is calculated from the image such that it takes minimum values at the point of interest like edges and boundaries. For example, for a grayscale image $I(x, y)$, the external energy can be calculated as

$$E_{ext} = -|\nabla I(x, y)|^2. \tag{2.18}$$


(a)

(b)

Figure 2.10: A snake model in a binary image (a) calculated GVF showing the balance of GVF at the edges; (b) movement of a snake towards an object, green line: initial contour.

A major drawback of the classic snake model is that the snake cannot move towards the objects that are too far. Many methods were purposed to solve this problem, one of the significant ones is Gradient Vector Flow(GVF) snakes purposed by Chenyang and

Jerry (1997). In this work, an original potential external force field was replaced by a GVF field. GVF field points towards the object boundary and varies smoothly over homogeneous regions using a computational diffusion process. GVF calculated for a binary image is illustrated in Figure 2.10(a), a crossing of the arrows at the edges is visible. A corresponding snake movement is shown in Figure 2.10(b), a green curve with an asterisk is an initial contour given by a user. In Caselles et al. (1993), a new model for active contours based on a geometric PDE was purposed. The model is intrinsic, stable and permits a rigorous mathematical analysis. It enables us to extract smooth shapes and it can be adapted to find several contours simultaneously. Some other noticeable works done regarding snake models are Cohen and Cohen (1993); Caselles et al. (1997); Chan and Vese (1999).

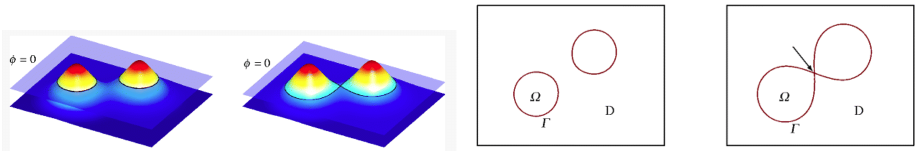


Figure 2.11: An example of level set function in two different time (van Dijk et al. (2013)).

Most existing active contour models are formulated on closed curves, while much fewer open active models are put to use. The open contour model has been adapted occasionally for applications like satellite imagery (Della Rocca et al., 2004), medical (Cohen, 1991), and road images (Cohen, 1996)(Melonakos et al., 2008). Apart from distinct energy functionals that suit their respective applications, the difference in open active contour can be characterized by their different boundary conditions. In fixed boundary, the end points are assumed to be known apiori with full certainty and need not shift during evolution. After the boundary points are set, rest of the snake evolution resembles the classical closed snake. Some of the methods introduced to reduce sensitivity to local minima during the curve evolution can be found in (Cohen, 1991) (Melonakos et al., 2008).

In some application, where the end points of the snake cannot be fixed at one point, the end points could be moved along the direction tangent to the curve (Hongsheng et al., 2009). A new class of open active contours with free boundary conditions, in which the end points of the open active curve are restricted to lie on two parametric boundary curves is suggested in Shemesh and Ben-Shahar (2011).

Another widely studied PDE based method for edge detection is the **level set method** purposed by Osher and Sethian (1988). The method is based on defining a contour V as a level set of a high dimensional function. The main advantage of level set over active contour is its ability to deal with the change of topology in the image as demonstrated in Figure 2.11. The level set method is more suitable for topology changes, which is appealing in situations in which the number of objects that must be detected is not known in advance. However, this makes the method less robust to noise because contours can over split. A simple representation of an interface Γ in the level set defined in Osher and Sethian (1988) is,

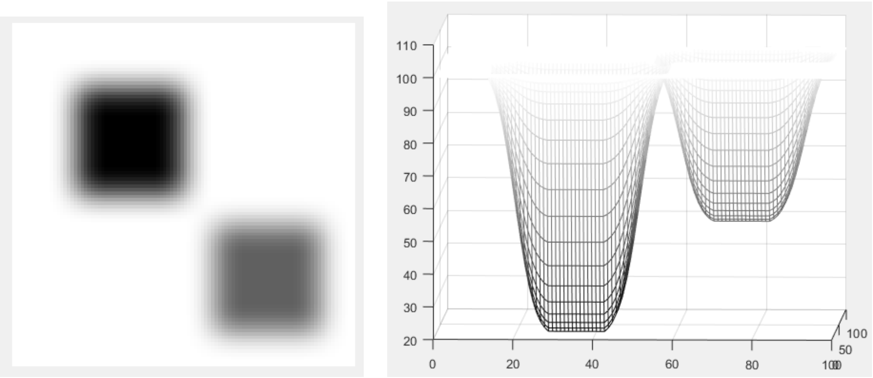
$$\begin{aligned} \phi(X) &= -d \quad \text{for } X \in \Omega^- && (2.19) \\ \phi(X) &= +d \quad \text{for } X \in \Omega^+ && (2.20) \\ \phi(X) &= 0 \quad \text{for } X \in \Gamma, && (2.21) \end{aligned}$$

where d is the euclidean distance to Γ. A new model for active contours based on; curve evolution, Mumford-Shah functional (Mumford and Shah, 1989) and the level set was purposed by Chan and Vese (1999). For a recent survey of level set snakes, see Gibou et al. (2017).

2.2.4 Region based techniques

Region based methods are generally used for image segmentation and object detection. Nevertheless, there is a close overlap between these methods and edge detection. For example in Figure 2.11, a level set approach detects the edges while also segmenting the objects from a background.

Split and merge is based on the divide and conquer approach. In this method, an input image is divided into sub regions until the sub regions become small enough for segmentation. Then appropriate merge rule is used to produce final segmentation results. This process is divided into four phases; split the image, merge similar sub regions, spatially adjacent regions and elimination of small regions. The criterion for the test is the homogeneity of the region, which can be grayscale intensity, mean, variance, etc (Gonzalez and Woods, 2000).



(a) (b)

Figure 2.12: An example of the watershed algorithm (a) a grayscale image; (b) representation of the grayscale pixel values as a topological field.

Watershed algorithm is based on the topology of the image and came from the field of mathematical morphology. Beucher and Lantuejoul were the first ones to apply the concept of watershed and divide lines to a segmentation problem in Beucher and Bilodeau (1979). The approach is simple, every pixel value is directly proportional to the height; such that the minimum is the deepest as demonstrated in Figure 2.12. The algorithm first divides the whole image into several catchment basins corresponding to its local minima and then start flooding the basins from the bottom. Starting from the minima of the lowest altitude, the water will progressively fill up the different catchment basins. Now, at each position where the water coming from two different minima would merge, the so-called dam/watershed line is built to prevent the merging of water from two basins, which eventually form the contours separating each catchment basins. The operation is demonstrated in Figure 2.13.

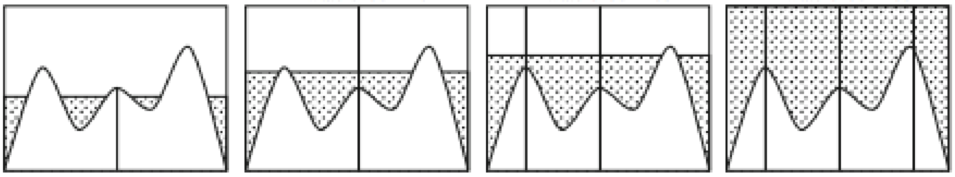
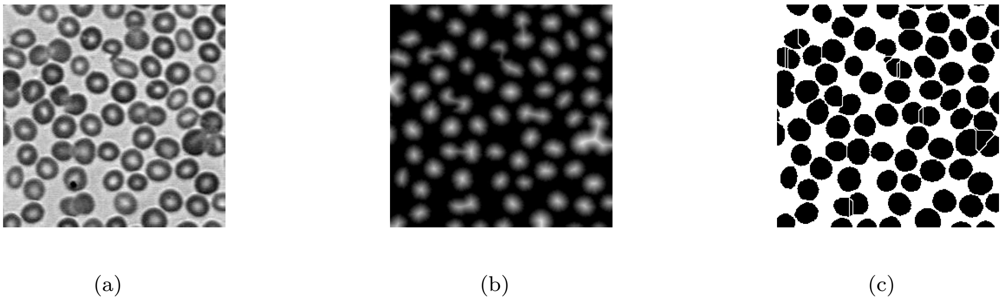


Figure 2.13: Watershed with immersion algorithm (Romero-Záliz and Reinoso-Gordo, 2018).



(a)

(b)

(c)

Figure 2.14: An example of the watershed algorithm (a) a grayscale image[1]; (b) a distanced transformed image; (c) final result of segmentation in thresholded image marker controlled technique.

The original algorithm has few drawbacks, like over segmentation, sensitivity to noise, poor detection of objects with low contrast and thin structure. To overcome the problem of over segmentation, a strategy known as marker-controlled segmentation purposed in Meyer and Beucher (1990). In Meyer (1994), the flooding process by using a distance transformed image was implemented, which still today is one of the best ways to implement the watershed algorithm. The distanced transform image represents the shortest distance between the pixel and the closest boundary. It is operated mostly on binary images (Kimmel et al., 1996). An example of implementing the marker controlled watershed algorithm and flooding in distanced transformed image is presented in Figure 2.14. Some of the other important contributions in the field of watershed segmentation are in

Vincent and Soille (1991); Beucher and Meyer (1993); Bieniek and Moga (2000); Bieniecki (2004).

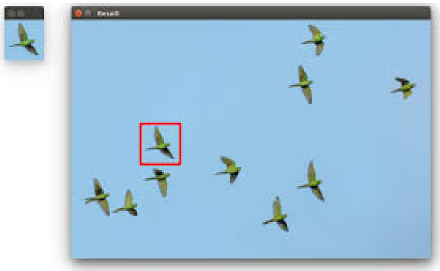


Figure 2.15: An example of template matching [2].

Template matching or pattern matching is one of the oldest image processing method, to detect a relatively smaller object in an image. It starts with creating a template of a relatively smaller size, whose one or multiple features match with the features of the desired object. Then, the created template is slid in a pixel-by-pixel basis, computing the similarity between the template features and its footprint in the image (Brunelli, 2009). An example of object detection using a template is illustrated in Figure 2.15. Some common features that are used for calculating a similarity while matching are, the sum of absolute difference (SAD), cross correlation, normalized cross correlation, the sum of squared error, mean square error (MSE), eigenvalues (Ouyang et al., 2012; Mahalakshmi et al., 2012).

2.2.5 Machine learning

Most of the work on edge detection in the field of machine learning makes a heavy use of the ground truth provided by the Berkeley Segmentation Data Set (BSDS) in Arbelaez et al. (2011), where each of the 500 images was processed by multiple human annotators. In the pioneering work Konishi et al. (2003), edge detection is formulated as a discrimination task specified by a likelihood ratio tested on the filter responses. Martin et al. (2004) carefully design features to characteristic changes in brightness, color, and texture associated with natural boundaries, and learn a classifier to combine the features. In Kivinen et al. (2014), the algorithm is divided into two parts; the first performs feature extraction by unsupervised feature learning techniques, while the second uses the features for edge prediction. The other one is proposed in Ganin and Lempitsky (2015), in which feature for image patch is learned using a conventional CNN and then the feature is mapped to an annotation edge map using kd-tree. Besides of supervised learning, Arbelaez et al. (2011),

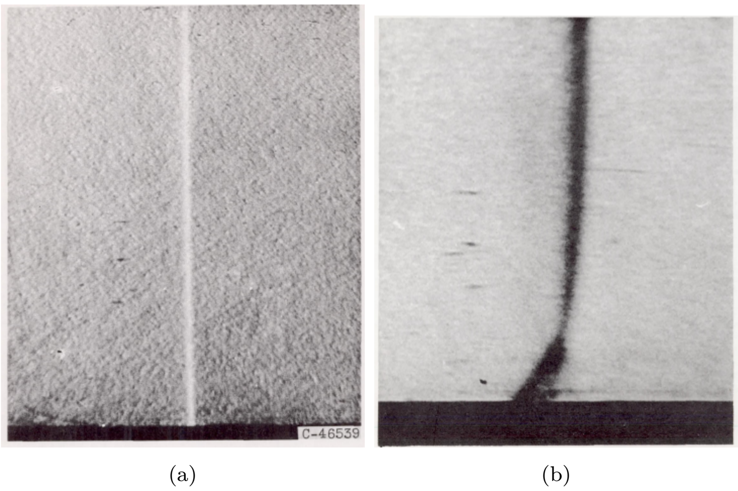
[1] Image is from lecture note: https://www.uio.no/studier/emner/matnat/ifi/INF4300/h11/undervisningsmateriale/INF4300-2011-f04-segmentation.pdf

[2] https://pythonspot.com/tag/template-matching.

combine multiple local cues into a globalization framework based on spectral clustering for edge detection.

2.3 Image processing in shock wave analysis

After Ernst Mach showed the existence of shock waves in a supersonic flow by capturing them in the high speed video, images slowly started to become a small but important part of the shock wave study. They provided the much needed visual verification of the various wave phenomena. In Mark (1958), which is the first of its kind to study about the SWBLI, images were extensively used to visualize the shock structure at various conditions. The images were captured using schlieren and shadowgraph visualization techniques, which are the most common technique for any kind of supersonic flow visualization. Two of the images from the memorandum is presented in Figure 2.16.



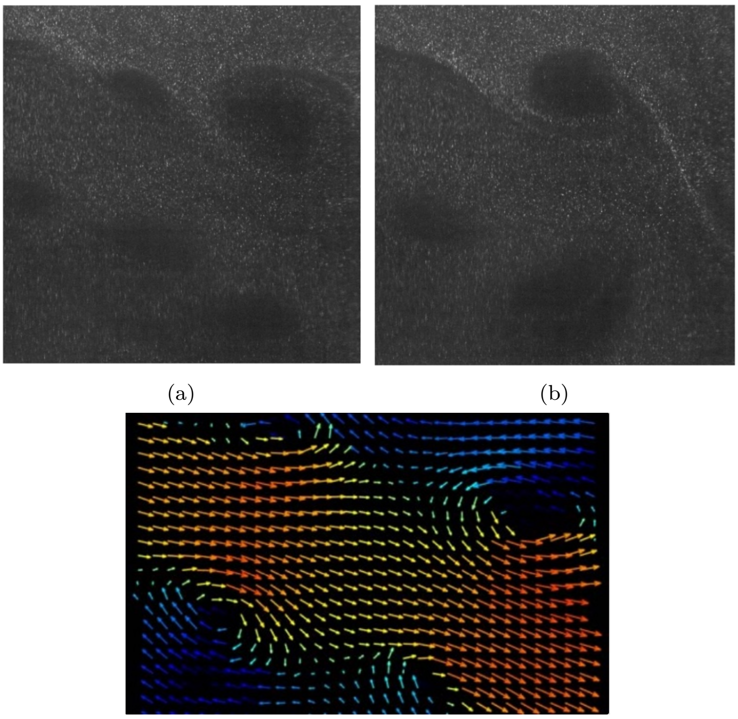
(a)

(b)

Figure 2.16: Images from Mark (1958); (a) Initial shock ; (b) reflected shock.

Flow visualization techniques during the last few decades have evolved massively. These techniques are not limited to just capturing a phenomenon now but to simultaneously diagnose the phenomena as well. Imaging based flow diagnostic techniques, like Particle image velocimetry (PIV) (Brossard et al., 2009), Background Oriented Schlieren (BOS) (Raffel, 2015), and Schlieren Image Velocimetry (SIV) (Biswas and Qiao, 2007) are gaining popularity as a novel measurement technique. PIV and SIV capture the entire 2D/3D velocity field by measuring the displacements of numerous small particles that follow the motion of the flow. The consecutive images are then cross-correlated to yield an instantaneous flow velocity field. An example of an instantaneous velocity field estimated by using PIV is presented in Figure 2.17 [3].

[3] https://www.cavitar.com/library/time-resolved-piv-measurements-cavilux-hf-diode-laser/



(a)

(b)

(c)

Figure 2.17: PIV (a),(b) Example of particle image; (c) an instantaneous flow field.

BOS on the other hand uses random dot pattern in the background and correlate them similar to PIV. Recently, NASA has successfully tested an advanced Air-to-Air BOS or AirBOS, in flight, capturing the first ever images of the interaction of shock waves from two supersonic aircraft in flight is shown in Figure 2.18[4]. The application of BOS to estimate displacement field of a laser-induced underwater shock wave can be read in Yamamoto et al. (2015). Development of schlieren and shadowgraph high speed imaging in the field of shock wave visualization is nicely reviewed in Settle and Hargather (2017). Furthermore, the imaging technique of color schlieren visualization to have a closer look at the problems associated with the shock bifurcation in a boundary layer is infallibly described in Kleine et al. (1992) Kleine (2005). Even though the methods of estimating flow velocity field is extremely popular in supersonic flow study, the techniques are based in a flow visualization technique and measurements rather than the actual image processing techniques.

On the other hand, the commercial computing software platforms like MATLAB, Python have developed their image processing techniques extensively. Thus, the quantitative analysis of schlieren and shadowgraph high speed videos using these platforms started to take the momentum from the past few years. However, there are just a handful of articles dedicated to image processing techniques for tracking shock wave front and most of them

[4] https://www.nasa.gov/centers/armstrong/features/supersonic-shockwave-interaction.html

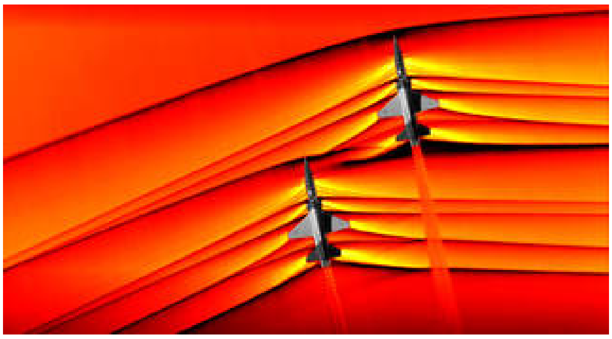


Figure 2.18: The images, originally monochromatic and shown here as colorized composite images, were captured during a supersonic flight series flown, in part, to better understand how shocks interact with aircraft plumes, as well as with each other.

are based on the simple inbuilt edge detection techniques, like Canny, Sobel, etc.

In Cui et al. (2013), the result of shock front wave tracking from different inbuild edge detection techniques in MATLAB was presented and discussed. The overall process contains four steps. The first two steps, smoothing, and enhancement are for removing noise and enhance the edges. The last two steps, detection, and localization are to determine the exact location of the edges. The challenge faced during the work however is dealing with the large volume of data captured by high speed camera and the high computational complex image processing algorithms. Similar work can be also seen in Guoshuai et al. (2019), the process includes background image subtraction, object area restoration, filtering, thresholding, and edge detection. In the end, the velocity of the shock wave is calculated using the detected instantaneous locations of shock waves. The result shows that image noises are eliminated effectively by background image subtraction in the frequency domain and proper filter method.

In Damazo (2013), a separate edge detection algorithm was developed and implemented. The mean transverse grayscale value was determined as a function of distance from the end wall by taking a vertical average of the image intensity. An example of determining the wave location is shown in Figure 2.19. The image processing part is not the focus of the work; however, it serves as a tool to validate experimental and simulation results. The process of comparing the results between image processing, numerical simulations, and experimental results continued for a very long time and still very much practiced (Kleine et al., 1992)(Akbar, 1997)(Timmerman, 2008). It is also worth mentioning that in most of the past work, the framerate of high speed video is lower than the framerate of high speed videos processed during this thesis.

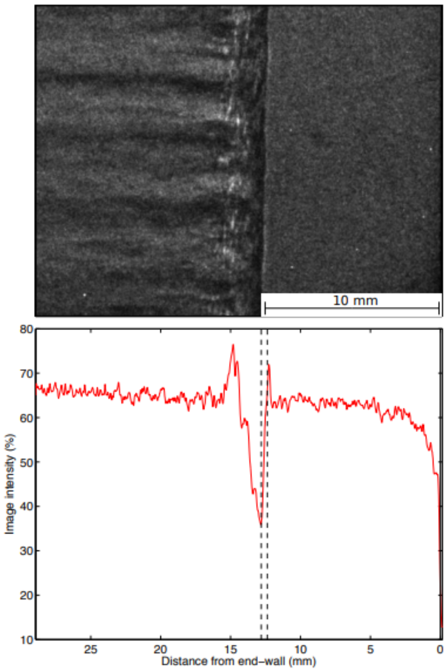


Figure 2.19: Schlieren image of incident detonation for shot 2152 with vertically averaged image intensity and determined wave location window. The solid black line shows the location of the end-wall and the dashed black lines represent the location of the detonation with uncertainty

2.4 Object tracking techniques

Even though wave tracking is not studied much, object tracking itself is a very well studied field. In the literature, there is numerous object tracking methods depending on its application like video surveillance (Ojha and Sakhare, 2015), traffic management (Kilger, 1992) (Ebbecke et al., 1997), etc. These methods implement relatively more advanced edge detection techniques such that they can track objects even in challenging situations. Nevertheless, the quality of results highly depends on apriori information about the object to be tracked, its features, and also on the quality of the video. Most of the algorithms work on the three steps process; object representation, feature selection, and object tracking. Any object can be represented in multiple ways, for example, points, contour, skeleton, template, etc. Some of the popular features that are used for tracking are color, edges, texture, etc. The actual object tracking procedures differ depending on the nature of the object representation and feature selected. For example, a point tracking fits for point representation and contour tracking for contour representation. For a detailed review of object tracking methods, please refer to Meenakshi and Gomathy (2015) Balaji and Karthikeyan (2017), while some of the tracking techniques which are in the scope of this thesis are described shortly in the following subsections.

2.4.1 Contour Based Tracking

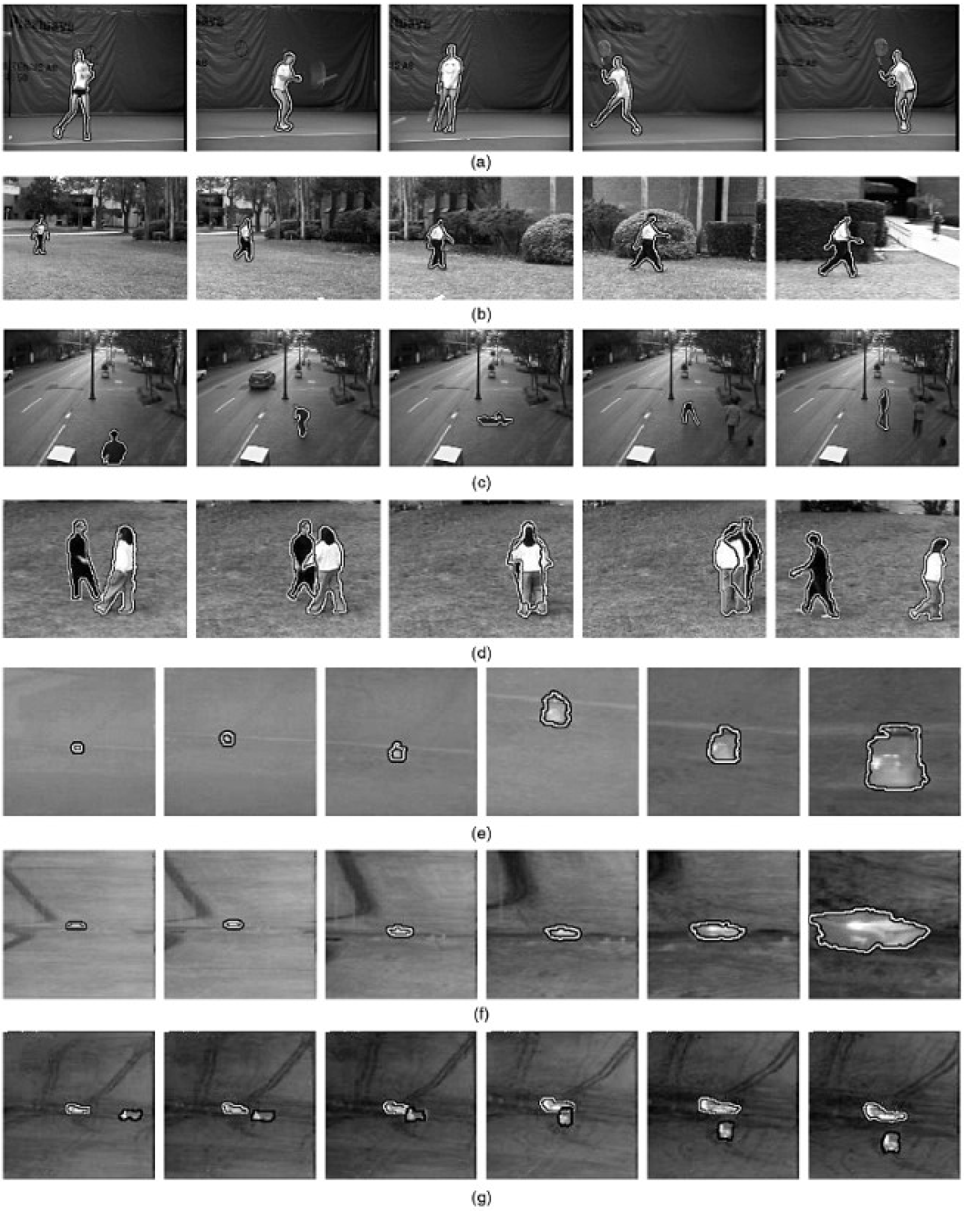


Figure 2.20: A snippet from Yilmaz et al. (2004) showing the contour based tracking of various objects.

Contour based tracking method tracks moving objects by representing them by their boundary contours. The method takes a contour from the previous image and iteratively progress it to fit into the object in the current image. Thus, it is required that at least some part of the object region in the previous frame is overlapped by the object region in the current frame. In Dokladal et al. (2004), closed active contours and a gradient based attraction field (described in Section 2.2.3) is used for tracking objects like people's face. A feature weighted gradient was used to estimate a gradient field and the attraction field is generated by a bi-directional, data dependent integrator. A faster active contour algorithm to reduce computation time and improve the performance of detecting and tracking the contour is proposed in Lee et al. (2009).

Using contour as representation makes it easier to represent complex and/or non-rigid objects and flexible in object representation (Patel and Patel, 2012). However, the method faces a huge challenge in handling occlusion. In Yilmaz et al. (2004), shape priors consisting of shape level sets are used to recover the missing object regions during occlusion. A snippet from Yilmaz et al. (2004) showing the contour based tracking of various objects is presented in Figure 2.20. Some of the other approaches to implementing contour tracking are state space models (Pu2, 2011), optic flow (Kanagamalliga and Vasuki, 2018), etc.

2.4.2 Region Based Tracking

Region based tracking method first segments an object region as foreground and the remaining as the background. For segmentation, at first, a background image is formed and the simplest way of forming a background image is perhaps taking an image of the scene when no objects are present. Then, based upon background subtraction, the foreground is typically identified by subtracting the background from the current image. The resulting image is then converted into a foreground and background (binary image) by using thresholding, i.e. if the value of the resulting pixel is less than the threshold it is assumed to be the background. The object region can be then tracked using approaches like cross-correlation function (Stephen et al., 2009). This method is used in the study of a real-time traffic monitoring system for the detection of moving vehicles (Kilger, 1992). However, the algorithm experienced difficultly to detect the vehicles under congested traffic, because vehicles partly occlude with one another.

In order to overcome the occlusion problem, an algorithm motivated by the assumption that the distribution of the error histogram of the occlusion region is different from that of the non-occlusion region is purposed in Song and Lee (2004). The proposed algorithm uses the mean and variance values to decide whether an occlusion has occurred in the region. Therefore, the proposed occlusion detection and motion estimation scheme detects the moving regions and estimates the new motion vector, while avoiding misdetection caused by the occlusion problem.

Not only vehicles, but a region based object tracking is also widely used for tracking pedestrians in a road. However, to segment, a moving human being in real-time is a very challenging task. In Wren et al. (1997), a real-time system called pfinder (person finder) was developed that solves the problem of tracking a single person using a fixed camera. The algorithm combines color and gradient information obtained from background subtraction to deal with shadow and unpredictable color clues. The obtained color clues are used to distinguish between the objects during occlusion. Instead of using a human body method, Mckenna et al. (2000) presents a color based tracking system. The visual parts depend on the clothing of a person and vary with a person's dress color. There are some situations like two humans have dressed similarly then the tracker will fail to track when they form a group.

2.4.3 Template based Tracking

Template based tracking method is relatively simple and a straight forward method to track objects. In the case of tracking a single object, a basic template matching can be used as described in Section 2.2.4 with different features like color or image intensity to form the template. The static template is formed from a reference image and runs over all the other images, whereas the dynamic/adaptive template changes its feature dynamically. An object tracking using an adaptive template matching algorithm by using the Sum of Squared Difference (SSD) as a matching parameter is presented in Chantara et al. (2015).

A tracking algorithm based on a multi feature joint sparse representation and adaptive templates to track multiple objects is presented in Hu et al. (2015). The variance ratio has been introduced to adapt the weights of different features. The template set has been updated adaptively using the tracking results. By using a sparse weight constraint, a large number of templates can be kept in the template set. Another algorithm suggested for tracking multiple objects is in Hai et al. (2002), which considers the image as a set of layers, and the number of layers is equal to the number of objects including an additional background layer. Features like layer appearance and a motion model corresponding to the object being represented define each layer. The background layer is used to compensate for any background motion so that an object's motion can be calculated from the compensated image. This way occlusion can be explicitly handled.

Another suggested method to detect occlusions uses the Bayesian decision theory (Zhou et al., 2006). This is done by using color intensity and color histograms as a feature representation and a similarity score for each detected pair. If the matching score is higher than a certain selected threshold value the pair is considered to be a match, and the templates used for tracking are updated. If the score is lower than the threshold value the object is further investigated to see if occlusion has occurred.

2.4.4 Machine learning

In recent years, with the rise of Deep Learning, the tracking algorithms have benefited from the representational power of deep models. In Kaiheng et al. (2019), a fast and accurate deep network based object tracking method is proposed. The method combines feature representation, template tracking, and foreground detection for robust tracking. In Supreeth and Patil (2018), Gaussian mixture model based object detection along with deep learning neural network is proposed. This model claims to handle false detections by improving the efficiency. A comprehensive survey on works that employ deep learning models to solve the task of multiple object detection on single camera videos is presented in Ciaparrone et al. (2020).

3 Materials

The details about experimental setups, high speed imaging and CFD simulation method are explained in this chapter.

3.1 Experimental setup

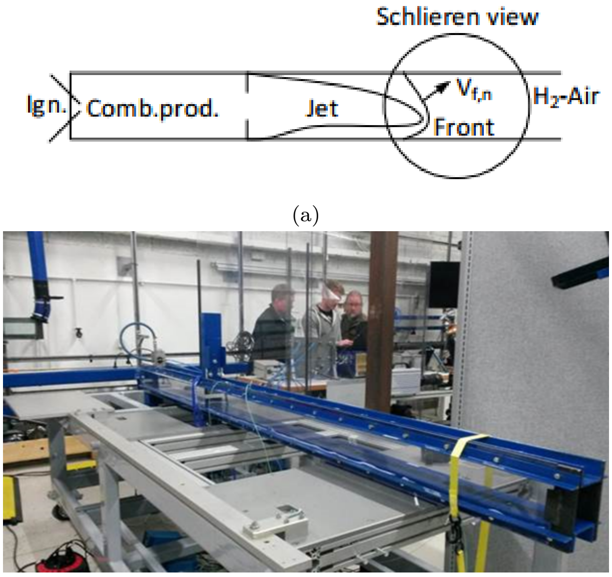
The high speed videos processed in this PhD study were recorded in two different experimental setups. The first experimental setup, USN detonation tube, is situated at the Process Technology Laboratory, USN. The second, GALCIT detonation tube (GDT) is located at the Graduate Aeronautical Laboratories California Institute of Technology, Caltech. Both of the experimental setups are shortly described in the following subsections.

3.1.1 USN detonation tube

A schematic representation of USN detonation tube and a photograph of the tube in an actual scene are presented in Figure 3.1. A three meter long rectangular channel is made up of the transparent polycarbonate walls, whose one end is closed and another is open to the atmosphere. An adjustable baffle type obstacle was located 1 meter from the closed end to generate the detonation wave. Five Kistler 603b type pressure transducers were used to record the pressure at various locations along the top and bottom walls. While conducting experiments, the channel was filled with combustible gases and ignited at the closed end. The gas filling was conducted by using two rotatometers, one for fuel and one for air. The two flows were set to a concentration and mixed in a T-junction before the mixture flowed into the channel. The details of the setup can be found in Haglund (2015).

3.1.2 GALCIT detonation tube

The GDT is a 7.6 m long, 280 mm inner-diameter detonation tube equipped with a 152.4 mm wide test section and two quartz windows to provide optical access. The possibility



(a)

(b)

Figure 3.1: (a) A schematic representation of the USN detonation tube (Bjerketvedt et al. (2015)); (b) a photograph of the USN detonation tube in the laboratory.

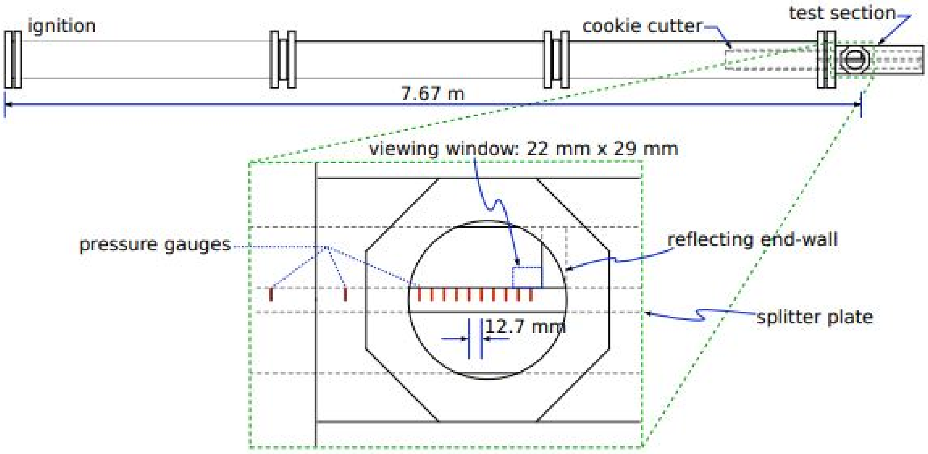


Figure 3.2: GALCIT detonation tube with test-section detail (Damazo, 2013).

of shock wave boundary layer interaction motivated the design and 68 construction of a splitter plate that raised the effective floor of the test section to the center of the windows. This allowed any interaction of the shock wave with the boundary layer to be observed. The geometry of the GDT, test section and splitter plate is illustrated in Figure 3.2. The test section has a rectangular cross section; this differs from the geometry used in the driven-thin detonation experiments in which the cross section was circular. However, considering the radius of curvature of the driven-thin specimen tubes of 63.5 mm was much larger than the expected boundary layer thickness and detonation cell size, we expect the

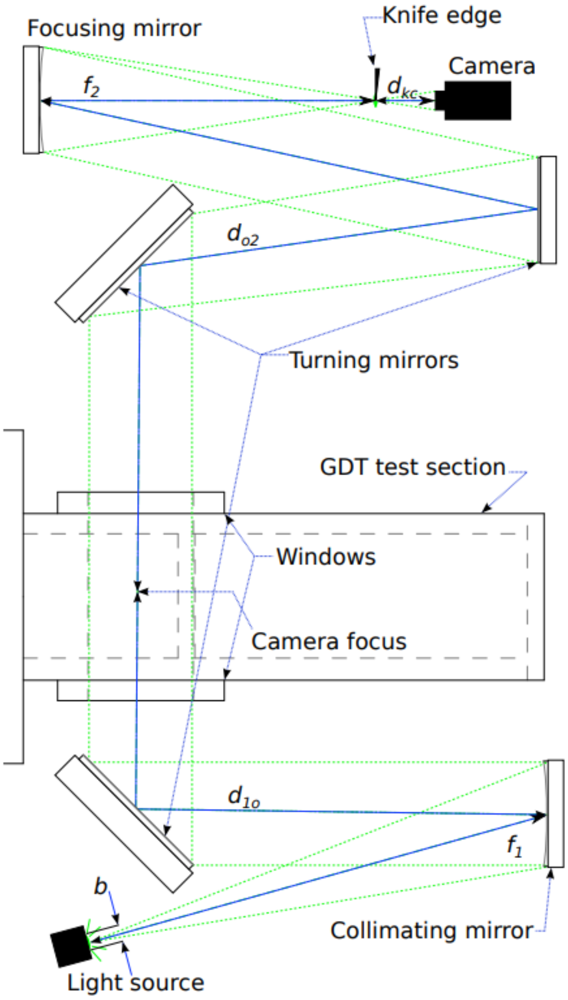


Figure 3.3: Representative schematic of schlieren visualization system as viewed from above in GDT (Damazo (2013)).

general flow features to be similar in the two experiments. The details of the setup can be found in Damazo (2013).

3.2 Shadowgraph and Schlieren setup

Two traditional flow visualization techniques for flows with variable density i.e. shadowgraph and schlieren methods have been used to gain qualitative insight into flow phenomena. Both imaging techniques translate the phase speed difference of a light passing through the medium, into the different intensities in a viewing plane (image). The phase speed difference is described by the refractive index of the medium, which is defined as,

$$\eta = \frac{c_0}{c}. \tag{3.1}$$

Here, c_0 and c are the speed of light in the vacuum and medium respectively. For gases, the refractive index is linearly dependent on the gas density, thus enabling the viewing of gas flow with the changing density (compressible flow). The angular deflection of the light ray travelling in z-direction when it passes through a region of varying density is given by,

$$\eta = K\rho + 1. \tag{3.2}$$

where K is Gladstone-Dale constant and ρ is the density and their product is much smaller than 1 which requires a very sensitive optics to detect the changes in density. Equation 3.1 also implies that,

$$\varepsilon = \frac{1}{\eta} \int \frac{\partial \eta}{\partial x} \partial z. \tag{3.3}$$

Here, x is the direction orthogonal to the light ray propagation. The deflection of the light ray to the extent of the region of varying density in the z-direction is what made visible in the images by using shadowgraph and schlieren techniques. The schematic diagram of the schlieren visualization system as viewed from above in GDT is illustrated in Figure 3.3. For the detail description of shadowgraph and schlieren technologies, refer to Settle (2001).

3.3 Kirana high speed camera

Figure 3.4: A high speed camera.

A high speed camera manufactured especially for capturing the fast phenomena namely 'KIRANA'[1] shown in Figure 3.4 was used for capturing all the high speed videos in this

[1] https://www.specialised-imaging.com/products/video-cameras/kirana

thesis. The camera was operated with the frame rate of either 500,000 or 200,000 FPS.

3.4 High speed video

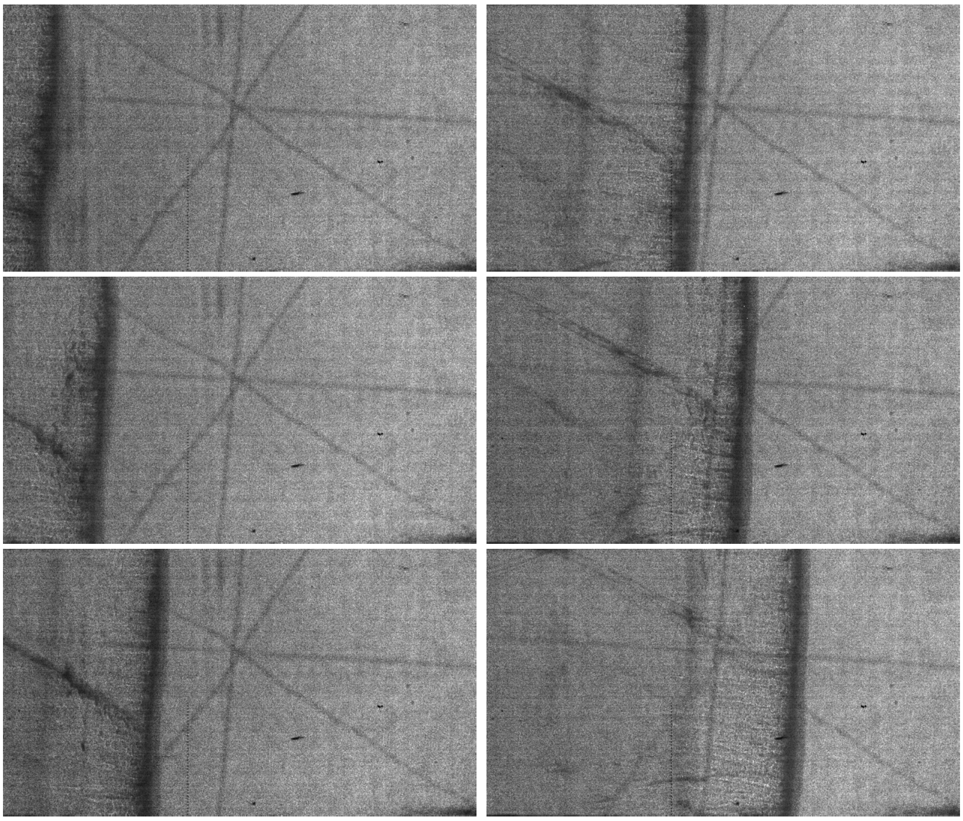


Figure 3.5: Some of the images from H_2 (Exp no. 00016) detonation experiments chronologically sorted from top to bottom, and left to right. The time between two subsequent frames is 24 μs.

During the initial days of this PhD, the experiments completed beforehand during the mater thesis Haglund (2015) were processed. These experiments were conducted in the USN detonation tube to study the detonation in stratified H_2 layer. Few images from a high speed video that shows detonation of H_2 gas in a benchmark experiment are presented in Figure 3.5. For uniformity, a test number in Haglund (2015) is written as an Exp no. in this thesis.

Later, SWBLI experiments were conducted for the study of a shock wave at GDT. Some of the images from one of the high speed video are presented in Figure 3.6 (another example is in Figure 1.8). When a planar incident shock wave propagating in a uniform gas in a shock tube (Glass and Patterson, 1955), hits the solid reflecting wall, a reflected shock

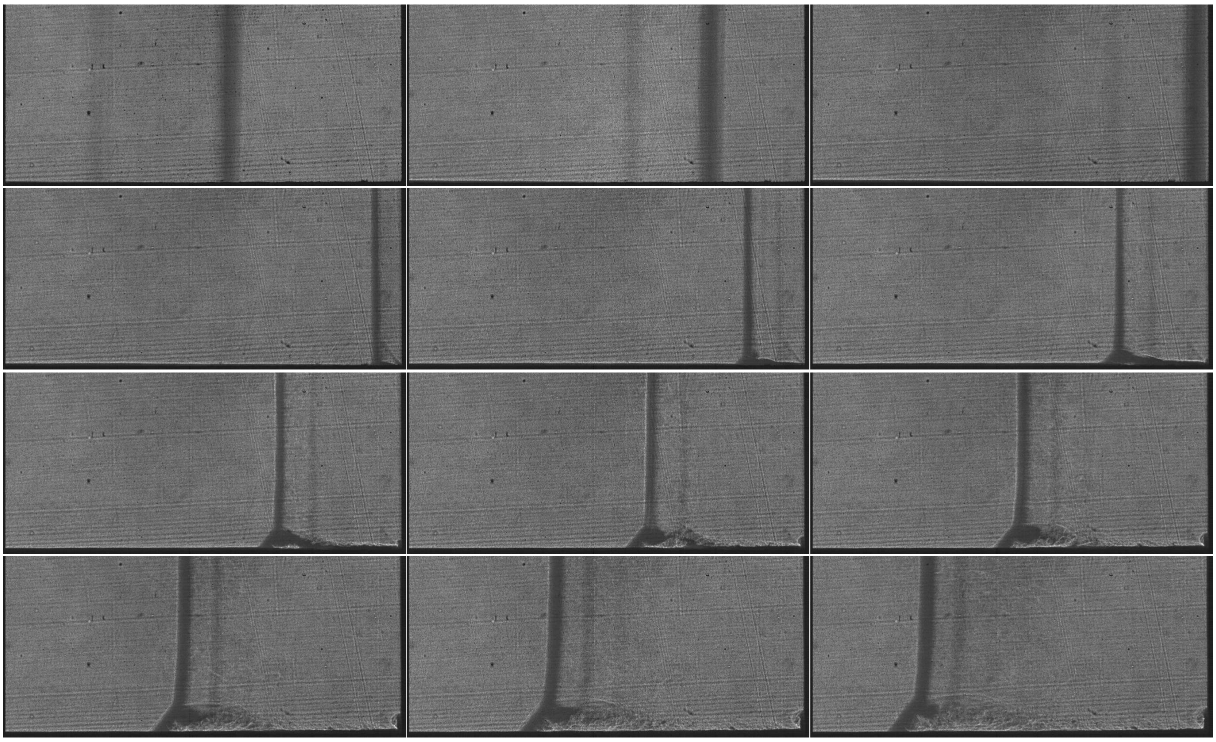


Figure 3.6: Images from a high speed video captured during SWBLI experiment conducted with 50 CO_2 + 50 Ar (Exp no. 2564) chronologically sorted from left to right, and top to bottom. The time between two subsequent frames is 24 μs.

wave initiated. The images in first row of Figure 3.6 visualize the incident shock wave propagating from left to right towards the end wall. The rest of the images in Figure 3.6 represent the reflected shock wave propagating away from the end wall. Immediately after initiation, it starts to interact with a boundary layer that was generated by the flow following the incident shock wave. The phenomenon is known as a shock wave boundary layer interaction (SWBLI) (Mark, 1958). In some cases, this interaction leads to a shock bifurcation. In this case, some part of the reflected shock around the boundary, precedes the main reflected shock, while a second shock (rear limb) develops behind the intersection of the preceding shock (oblique shock) and the main shock (normal shock). The phenomena can be observed in the images from below the first row in Figure 3.6. The meeting point of these three different shocks is what called a triple point. In other cases, the interaction can just lead up to a slightly distorted foot without a triple point structure (Babinsky and Harvey, 2011).

The images in both Figure 3.5 and Figure 3.6 were cropped to the appropriate size to avoid the unwanted portions like a viewing window. These high speed videos were captured at the frame rate of 500,000 FPS and the time between two subsequent images is 24 μs. Each high speed video consists of 180 images and has a size of $[768 \times 924]$ pixels. However, the size of the images in the metric system in each video might be different. The image size in the metric unit is either about [81 × 97] mm or [60 × 72] mm depending on the scaling factor of 9.528 or 12.902 pixels/mm respectively.

3.5 CFD simulation method

To compare the calculated shock speed with a simulation result, a physical process was simulated using an in-house CFD-code solving the mass, momentum and energy equation using the ideal gas law (Vaagsaether et al., 2007; Vaagsaether, 2010; Gaathaug et al., 2012). The equation set is solved by a centered flux limiter method (FLIC) for the hyperbolic part of the equations and central differencing for the viscous stresses. The 2-D simulation domain shown in Figure 3.7 is [70 × 15] mm with constant mesh size of 10 μm. The heat capacity ratio for CO_2 was set to 1.23 for this simulation. A normal propagating shock wave is placed 200 control volumes from the left boundary as the initial condition. The Mach number of the shock wave is set to 2.4. Few images of the reflected shock in the created domain are presented in Figure 3.8. Compare to the images from high speed films, these images are better quality in terms of background noise. For these images, the front position is determined at the first black pixel from left.

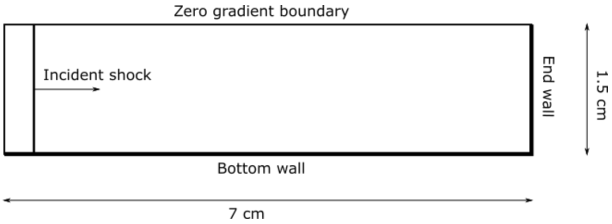


Figure 3.7: The 2-D simulation domain with a normal incident shock wave.

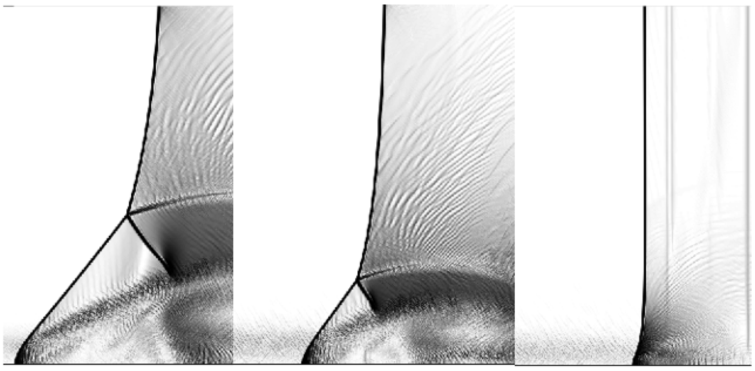


Figure 3.8: The images of the reflected shock generated during a CFD simulation for the same experiment setting.

4 Methodology

The image processing algorithms developed for image filtering, tracking a front position, and calculating wave properties like speed, position of a triple point, pressure, etc. are described in this chapter. The developed front tracking methods are based on some of the edge detection and object tracking methods described in Chapter 2.

As this PhD is application driven, the approach has been to develop algorithms that are straight forward, understandable, and at the same time efficient. Many of the object tracking methods are focused on tracking multiple objects in a real-time situation. Both of these requirements were not necessary for the problem in hand i.e. front tracking. Thus, the tracking methods developed are comparatively simpler than existing object tracking methods. This might encourage researchers to use image processing for wave tracking in the gas study. However, each high speed video differs in quality, meaning that basic edge detection techniques could be used for high quality videos, whereas advanced techniques had to be developed while processing low quality videos.

The development of front tracking began with a contour based method. The second method explored the region based segmentation to overcome the challenges faced during the contour based method, for example, initialization of the contour, parameter tuning, etc. The next approach is a template based tracking, which minimizes the tracking error and also simpler than the previous methods. The developed algorithms are described in the following subsections in accordance to the timeline of development. Due to the time limitations, the front tracking algorithm based on machine learning could not be fully developed. Though this work couldn't progress forward and isn't mature enough to draw any conclusion, it might be one of the interesting work in the near future. The brief introduction and preliminary results are therefore presented as unpublished work in Part III.

In this section, the images from the high speed videos in the figures are appropriately cropped to [400 × 400] for better visualization.

4.1 Image filtering

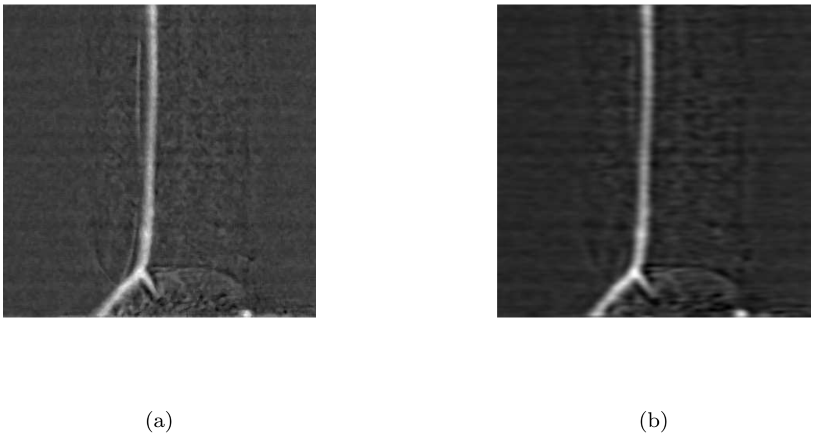
Noise and the other artifacts in the images of the high speed videos are mainly introduced by the ongoing chemical changes, filming technology, and the experimental equipments.

Two image filtering algorithms were developed to eliminate unwanted noise in the images, namely

1. Non-linear median filter in the spatial domain

2. Linear/convolution filter in the Fourier domain.

To make filtering more efficient, background subtraction was implemented before applying any of these filtering algorithm. Background subtraction is a popular pre-processing method for moving object detection in a video recorded with a common background (Piccardi, 2004). One way of forming an acceptable background image for a high speed video is by taking an average of all the images without a visual wave front. For those cases, where there is no image without a wave front, a single image from the video between the incident wave and the reflected wave is selected. In both ways, a background image was constructed with a prior study of the images. The constructed background image was then successively subtracted from all the images with a visual front.



(a)

(b)

Figure 4.1: Image filtering (a) median filtering corresponding to top right image from Figure 1.8 with [9 × 9] kernel size. (b) Frequency filtering corresponding to Figure 3.6 with cut off of 50 Hz.

The result of median filtering in a background subtracted image from a high speed video is shown in Figure 4.1(a). The median filter works adequately, however, when the images contains a higher amount of distortion, this filtering technique was relatively less efficient. Therefore, a faster and relatively better frequency filtering algorithm was developed. The result of a low pass filtering with a cut off frequency of 50 Hz is shown in Figure 4.1(b). In some cases, the resulting image after filtering is converted into a binary image by using thresholding, i.e. if the value of the resulting pixel is less than the threshold it is set to be 0 and more than threshold to be 1. An example of a binary image can be seen in Figure 4.5(a).

4.2 Open active contour model

The first algorithm developed to track the wave front was based on the active contour model introduced in Kass et al. (1988). As it is based on the evolution of the curve, any kind of wave could be tracked independently of shape and size. This is one of the major reasons to implement this technique for front tracking. An open contour model with a fixed boundary condition was developed to track the wave front from top to bottom boundary, an example can be seen in Figure 4.2. The traditional snake $V(s) = (x(s), y(s))$, described in Section 2.2.3, which aims to minimize the energy function E_{snake} (2.16) must satisfy the Euler equation,

$$\alpha V''(s) - \beta V''''(s) - \nabla E_{ext} = 0. \tag{4.1}$$

This can be viewed as a force balance equation,

$$F_{int} + F_{ext} = 0, \tag{4.2}$$

where,

$$F_{ext} = -\nabla E_{ext} \tag{4.3}$$

$$F_{int} = \alpha V''(s) - \beta V''''(s). \tag{4.4}$$

An GVF field described in Chenyang and Jerry (1997) was used as the external force F_{ext}. The GVF field $G(x,y) = (u(x,y), v(x,y))$ can be calculated by solving energy minimizing function,

$$\varepsilon = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |G - \nabla f|^2 dxdy. \tag{4.5}$$

Here $f(x,y)$ is an edge map derived from the image, having the property that is larger near the image edges, for example a gradient of an image.

By approximating the derivatives with the finite differences, the corresponding internal force F_{int} at any point V_i of the snake can be written in a matrix form as,

$$E_{int}(V_i) = A.V_p$$

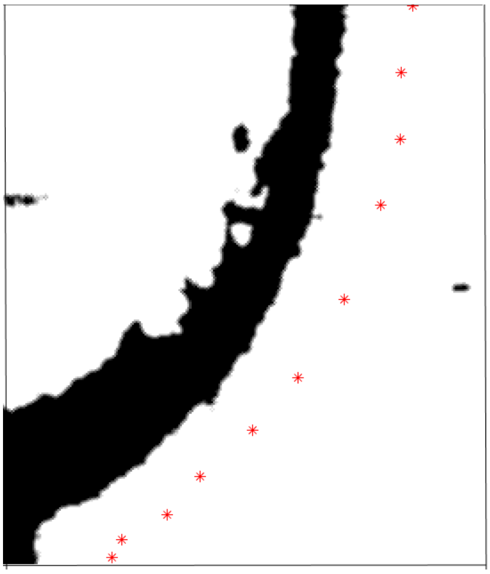


Figure 4.2: The initial points selected manually for initializing an open snake.

Here A is penta diagonal matrix and V_p is a vector of the 5 consecutive points of the contour V as,

$$A = \begin{bmatrix} \beta & 0 & 0 & 0 & 0 \\ 0 & \alpha - 3\,\beta & 0 & 0 & 0 \\ 0 & 0 & 2\,\alpha + 6\,\beta & 0 & 0 \\ 0 & 0 & 0 & -\alpha - 4\,\beta & 0 \\ 0 & 0 & 0 & 0 & \beta \end{bmatrix}$$

$$V_p = [V_{i-2} \quad V_{i-1} \quad V_i \quad V_{i+1} \quad V_{i+2}]^T.$$

The open contour method generally relies on a fixed boundary condition (Cohen, 1991) (Melonakos et al., 2008). Apart from boundary conditions, energy definition also changes accordingly. The few available open contour models are very challenging in terms of developing and implementing. They require adequate knowledge of computer vision, applied mathematics and image processing. So in this work, we tried to minimize the complexity as much as possible and stay close to the original concept of energy minimizing. A simple fixed boundary condition that the endpoints of the snake move along the boundary is created. If n be the number of points in the open contour such that $i = 0 : n$, in contrast to the closed snake $V(n) \neq V(0)$, i.e. all the elements in the matrix V_p doesn't exist for the two boundary points. So, while estimating the internal energy for these boundary points, the matrix V_p was changed accordingly considering only the points that exist. The initialization of the snake such that the endpoints lie as close to the boundary as possible can be seen in Figure 4.2. Taking advantage of apriori information about the image size, while interpolating the initial points, it was made sure that the two endpoints of the snake are always attached to the first and the last row.

While implementing the open snake model, the internal energy for the overall curve was estimated by using a pentadiagonal banded matrix (see Appendix of Kass et al. (1988)).

4.3 Watershed algorithm

The watershed algorithm is a region based method to segment the object from the background. The main cause of applying the watershed algorithm for the front tracking is to eliminate the need for any curve and challenges with curve progression to detect edges. Instead, the approach is to segment the front from the background precisely and detect edge by finding the position of the pixel separating the background and the foreground.

A sequential procedure of the watershed algorithm is explained below, before that two morphological operations that are used extensively for watershed segmentation are shortly introduced;

- Connected component: Any two pixels are said to connected component of a subset of pixels S, if there exists a path between them consisting entirely of pixels in S. For example there are two connected components in Figure 4.3(a) and one in Figure 4.3(b).
- Dilation: It is an operation carried away by a structuring element to expand the shape/boundaries present in the image.

As mention in Section 2.2.4, the watershed algorithm works better when applied into a distanced transformed image rather than the original ones. The stepwise procedure to obtain distance transformed image and watershed segmentation is: raw image from high speed videos → background subtraction →median filtering → Otsu's method of image thresholding (Otsu, 1979) → morphological dilation (thickening)→ morphological closing → distance transform → watershed segmentation.

How watershed algorithm works can be briefly explained with Figure 4.3. Let $d(x,y)$ be the input image (a distance transformed image) and M_1, M_2 be the coordinates of the regional minima. The flooding starts from $\mathrm{n} = min + 1$ to $\mathrm{n} = max + 1$ iteratively, where min and max is the minimum and maximum intensity of $d(x,y)$. Then let $C_{\mathrm{n}-1}(M_1)$ and $C_{\mathrm{n}-1}(M_2)$ be the set of coordinates of points in the catchment basin associated with M_1 and M_2, flooded at stage $\mathrm{n}-1$ as shown as two gray regions in Figure 4.3(a). Now moving to stage n Figure 4.3(b), it can be seen that two connected components in stage $\mathrm{n}-1$ (showed by dashed lines) becomes one connected component q. This also implies that the water between two catchment basins merged and there is need to build a dam.

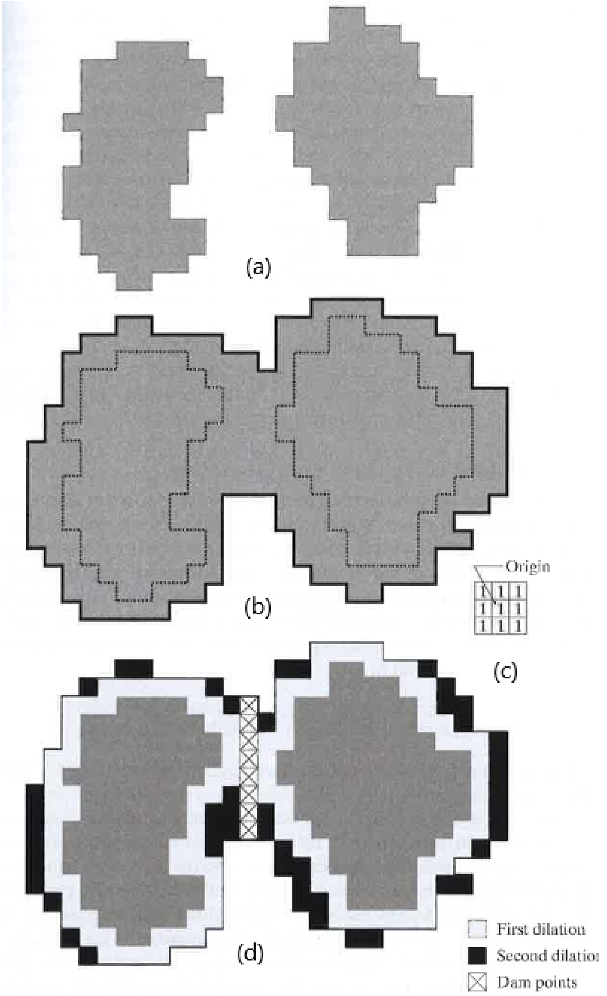


Figure 4.3: (a) Two partially flooded catchment basins at stage $\mathrm{n}-1$ of flooding; (b) flooding at stage n, showing that the water has spilled between two basins; (c) structuring element for dilation; (d) result of dilation and dam construction (Gonzalez and Woods, 2000).

Let $T[\mathrm{n}]$ be the set of coordinates of points in $d(x,y)$ lying below the plane $d(x,y) = \mathrm{n}$ and C_{n} denote the union of the flooded catchment basins, such that $C_{\mathrm{n}-1}$ is subset of C_{n} subset of $T[\mathrm{n}]$. The process of obtaining C_{n} from $C_{\mathrm{n}-1}$, is by dilation iteratively subjected to two conditions: 1) dilation has to be constrained to q and 2) dilation can not be performed on pixels that would cause two basins to be merged (form a single connected component). Suppose Q be the set of connected components in $T[\mathrm{n}]$, for each connected component q in Q, there are three possibilities:

- $q \cap C_{\mathrm{n}-1}$ is empty, new minima.
- $q \cap C_{\mathrm{n}-1}$ contains one connected component of $C_{\mathrm{n}-1}$, q lies in that catchment basin.

- $q \cap C_{n-1}$ contains more than one connected component of $C[n-1]$, q is then on a ridge between catchment basins, and a dam must be built to prevent overflow.

In Figure 4.3(b), the first possibility is not applicable as there is no other connected component that existing ones. The first dilation of C_{n-1} with structure shown in Figure 4.3(c) is shown by light gray colour in Figure 4.3(d). Each q component at this step agrees with the second condition expanding the boundary of each original connected component. In the second dilation (shown in black), several components fall into the third condition shown by the cross, so the dam is created at these components. The definition of the algorithm and developed algorithm is based on Gonzalez and Woods (2000) chapter 10. One of the results of the watershed algorithm is depicted in Figure 4.4. The position of the front is then detected by using apriori information of the wave propagation direction. In Figure 4.4, the wave is propagating from right → left, so a front position was picked at the first pixel from left between the background and the foreground.

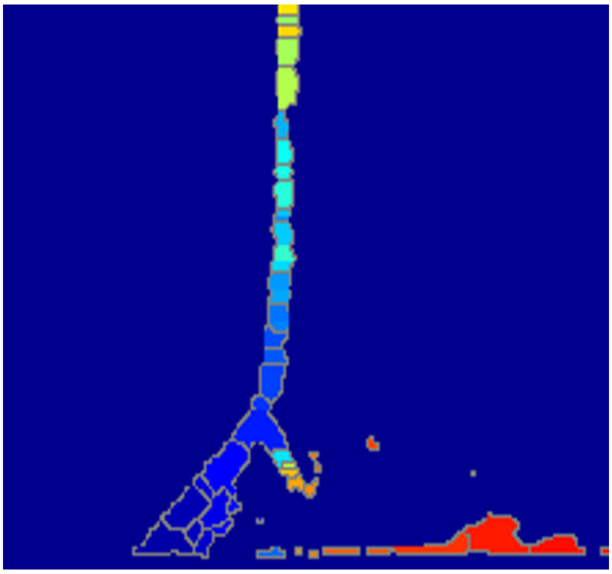
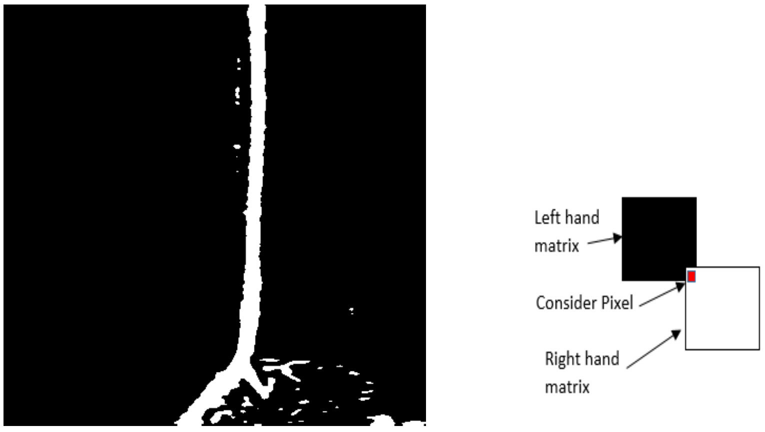


Figure 4.4: Image segmented by the watershed algorithm.

4.4 Template matching

Figure 4.5(a) is a segmented binary image formed by background subtraction → convolution filtering → image thresholding. As can be seen in Figure 4.5(a), there are few noises in front of the front, hence a basic approach of picking the first white pixel from the left will give few errors. Implementation of the watershed algorithm does remove most of the noise, however doesn't remove all. Thus, instead of going forward with the watershed algorithm and introducing more image morphological steps, a one step approach of template matching is used to find the position of the front.



(a) (b)

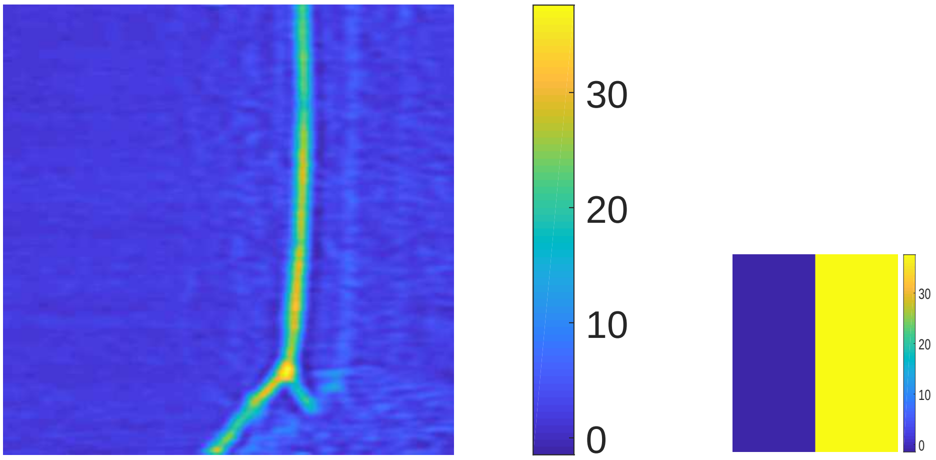
Figure 4.5: Binary template matching (a) segmented image; (b) template consists of two $[5 \times 5]$ matrices for a consider pixel.

A binary template which consists of two $[5 \times 5]$ matrix was created as in Figure 4.5(b). The matrices are 45 degrees skewed step like structure so that a single template can be used for tracking both normal (straight) and oblique (tilted) shock. The element values for the left hand matrix of the template are set to 0 (black) while for the right hand matrix to be 1 (white). The matching error is the sum of the absolute differences between the pixel values of the template and the footprint template created around the considered pixel in the image. Assuming template $\boldsymbol{T}$ of size $[a{\times}b]$ slides over an image $\boldsymbol{I}$, then at each position (m,n) in $\boldsymbol{I}$, SAD is estimated as in 4.6. The template matches best in the image pixel where SAD is the minimum.

$$SAD(m,n) = \sum_{k=1}^{a} \sum_{e=1}^{b} |I(m+k,n+e) - T(k,e)|. \tag{4.6}$$

Most of the previous work in the field of front tracking is done after the image threshold into a binary image (Guoshuai et al., 2019). However, image thresholding especially in the low contrast images mostly results with an error of few pixels around the actual edges (Villarrubia et al., 2002), which at the end might influence the calculated shock speed. Hence, a dynamic template matching approach is introduced for tracking front in grayscale images and eliminate image thresholding.

A template of pre-defined size of $[5 \times 20]$ was created, an example is presented in Figure 4.6(a). The values of the template depend on the minimum and maximum intensity of the considered image as in Figure 4.6(b). One half of the template contains the minimum intensity value which should technically be the intensity of background, while the other



(a)

(b)

Figure 4.6: Dynamic template matching (a) grayscale image; (b) $[5 \times 20]$ size template consists of maximum and minimum intensity value.

half contains the maximum intensity value which should be the intensity of the wave. Matching is based on MSE, which takes a mean of the squared difference between the intensity of each pixel in the template and the corresponding pixel in its footprint in the image. At each position (m,n) in I, MSE is estimated as in (4.7). The template matches the best in the image pixel where MSE is the minimum.

$$MSE(m,n) = \frac{1}{ab} \sum_{k=1}^{a} \sum_{e=1}^{b} [I(m+k, n+e) - T(k,e)]^2. \tag{4.7}$$

4.5 Segmented regression

Segmented regression is a method of fitting multiple lines or curves in a single data set, an example is presented in Figure 4.7. It is mainly useful when there are different patterns in the same dataset and the timing of change in the pattern is unknown. However, it is necessary to know the shape of the pattern, for examples if they are straight lines or higher degree polynomial curves etc, (Ryan and Laurie, 2007).

In order to calculate some of the primary information about a shock wave, it is important to separate a tracked front into normal shock and oblique shock (see Figure 1.9). By taking

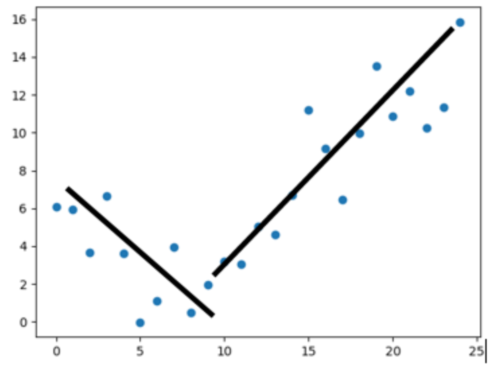


Figure 4.7: An example of segmented regression.

advantage of a prior knowledge that the both shocks are of first order polynomial lines, segmented regression method was implemented. For fitting two first order polynomial lines, it requires one break point $\boldsymbol{BP}$, and the model can be written as,

$$y = a_1 x + b_1 \quad \text{for} \quad x \leq BP \tag{4.8}$$

$$y = a_2 x + b_2 \quad \text{for} \quad x > BP. \tag{4.9}$$

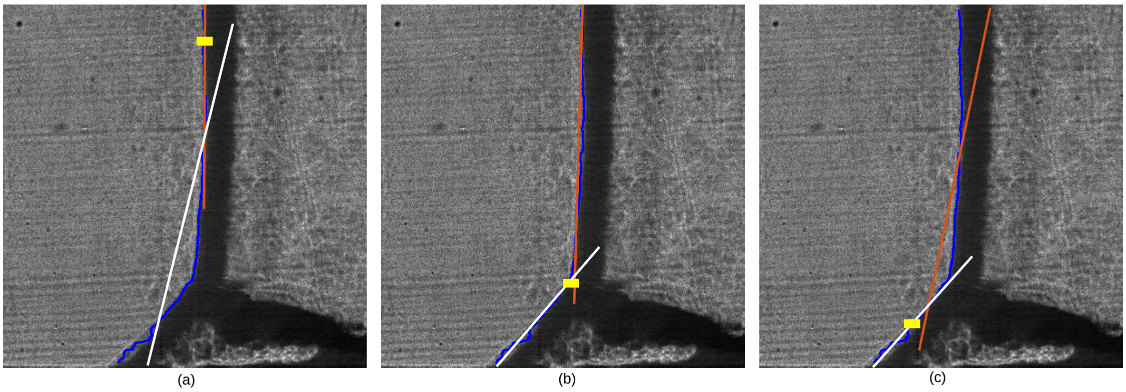


Figure 4.8: Segmented regression process. The yellow marks represent different $\boldsymbol{BP}$ for line fitting. The blue curve represents the tracked front. The orange line is a line fitted for the normal shock, while the white line gives the line fitted for the oblique shock.

An optimum breaking point $\boldsymbol{BP_o}$ was estimated by using a mean square error between the fitted lines and respective front points. Figure 4.8 summarizes the segmented regression process of determining the $\boldsymbol{BP_o}$ by consecutively fitting two lines to the front. The blue curve represents the tracked front and the yellow mark yields the breaking point($\boldsymbol{BP}$), while the orange line represents the first line fitted as in (4.8) (note the difference between image and (x,y) coordinate system). The white line is then fitted to the remaining points below $\boldsymbol{BP}$ as in (4.9). The process follows a brute force approach, from top to bottom, starting $\boldsymbol{BP}$ at row no. 10 and for each iteration, the $\boldsymbol{BP}$ moves down by one row. After each iteration, the least square error is calculated for both fitted lines with their respective points and the errors are summed up and stored. For example, in Figure 4.8(a) the orange line will give a small amount of error as it almost coincides with the blue curve above the yellow point, while the white line is misplaced and causes large errors.

Figure 4.8(b) shows the fitted line when the separating point is located near the triple point and thereby the fitted lines representing the normal and the oblique shock are rather accurate. The total error gathered from both lines seems to be at a minimum at this location. Further down, see Figure 4.8(c), the error increases. For each image, after the process was finished, the $\boldsymbol{BP}$ that gave the minimum error was considered as $\boldsymbol{BP_o}$ and also the separating point between the normal and the oblique shock.

4.6 Shock wave information

In this section, a method to calculate shock wave speed by using the front position tracked by image processing algorithms is presented. In addition, a short description of shock wave relationships, which were used to estimate shock properties like temperature and pressure are described shortly.

Shock speed

A two point method of estimating speed of any object is to calculate the distance travel by the object and divide by the time taken. Let X_1 and X_2 be the position of the shock front relative to the reflecting wall, at time t_1 and t_2 respectively, such that $t_2 > t_1$. The shock speed U can then be defined as a first order polynomial form,

$$U = \frac{X_2 - X_1}{t_2 - t_1} \;\; [m/s]. \tag{4.10}$$

Let's say position of fronts tracked by the image processing algorithm, at time t_i be x_i. The time t_i is extracted from the frame rate of the camera considering $t_0 = 0$ for the first front. Then shock speed U at time t_i, i.e. U_i is given by,

$$U_i = \frac{(x_{i+\mathrm{n}} - x_i)}{t_{i+\mathrm{n}} - t_i}, \tag{4.11}$$

where n is a time step.

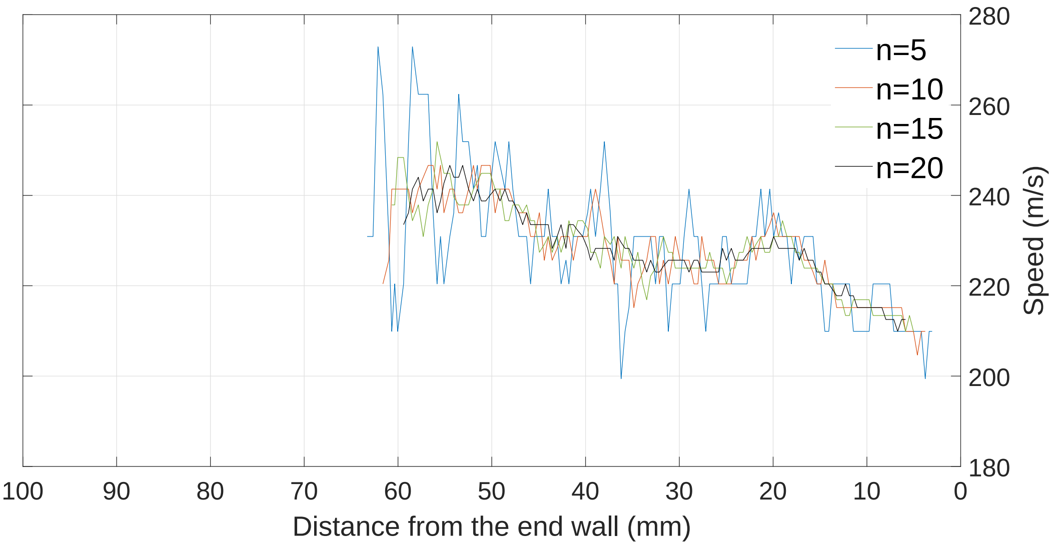


Figure 4.9: Normal shock speed calculated by using a two point method as in (4.11) with varying n (Exp. no 2516).

However, using the two point method for the positions tracked in the images gives high resolution and an unusual oscillating pattern in the calculated speed. One of the causes of high resolution is a high frame rate (i.e time) and low resolution in length scale ($\approx$ 0.1 mm/pixel). By increasing the time step, the resolution reduces, however the oscillation remains. The results can be observed in Figure 4.9. A similar kind of oscillation phenomenon was also noticed in (Timmerman, 2008). The detail about the error and sources of oscillations in the result can be read in (Settle and Hargather, 2017).

One way of reducing the resolution and oscillation is by using non-linear square fitting method (NLSFM). In Damazo (2013), NLSFM was used to determine an average speed of incident detonation using first order polynomial, and for the reflected shock a second order polynomial was used. However, because there were enough data points even in a small time interval, we choose to estimate shock speed in a piecewise manner rather than averaging over the incident and reflected time. As the process were piecewise the movement of shock was extremely small, therefore a first order polynomial was used for both the incident shock and the reflected shock.

Lets consider n time steps/frames to estimate the speed at each position, i.e. we assume the shock speed is constant for n time steps. From (4.10),

$$X_{i+1} = U \cdot (t_{i+1} - t_i) + X_i \quad i = 1 : \mathrm{n} + 1. \tag{4.12}$$

The points which are separated by equal distance i.e. constant speed are represented by blue circles in Figure 4.10. The y-axis values in this figure is arbitary used for better

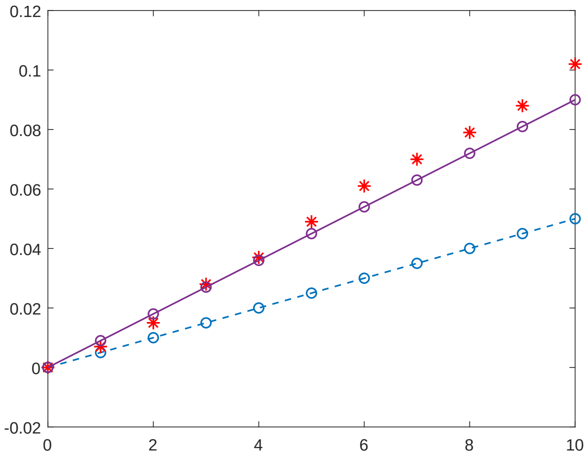


Figure 4.10: NLSFM, blue circles-calculate positions in different time instant with constant speed, red asterisks-actual front positions tracked by image processing, purple circles-fitting results of blue dotted line to minimize error between circles and corresponding red asterisks. The y-axis values in this figure are arbitrarily used for better visualization and don't relate to the actual wavefront speed.

visualization and doesn't relate to actual wave front speed. However, the actual tracked positions x_i didn't match with X_i. So, the basic NLSFM, fits the blue dotted line in such a way that error between corresponsing blue circles and red asterisks becomes minimum, for example the puprle line in Figure 4.10.

The error can then be defined as (4.13),

$$e = \text{sum}(X_i - x_i) \quad i = 1 : \text{n} + 1. \tag{4.13}$$

By using 'lsqnonlin' function in MATLAB, we find the optimum value of U that minimizes the error function e. The optimum value of U then gives the speed of the shock at position i. Note that, $(t_{i+1} - t_i)$ is constant as the camera was operated with a fixed frame rate so that there is no need to optimize with respect to t. The initial position of the front for (4.12), is considered the first tracked shock front i.e. $X_0 = x_0$. Similarly, the second iteration continues with frames between $i+1 : \text{n}+1$ and so on.

Shock relations

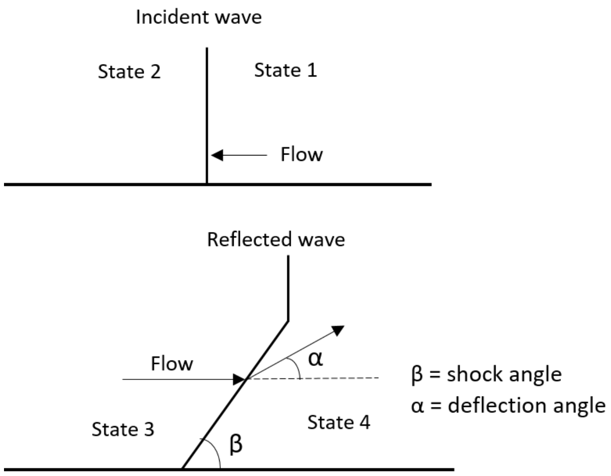


Figure 4.11: Different states during SWBLI experiment in stationary shock system.

By calculating a shock speed, other shock properties like pressure and temperature can be estimated using various shock relations. However, the shock speed calculated by the image processing framework is relative to the lab coordinate system i.e. shock is moving and flow is stationary. In order to use the shock relations and estimate shock properties, the coordinate system needs to be changed into a stationary shock system i.e. shock is stationary and flow is moving (Mark, 1958).

Without going into detail, some of the shock relations that were used or can be used for estimating temperature, pressure, and density at different stages of shock wave propagation are written below. For illustration of different states see Figure 4.11 and the nomenclature used are :

M = Mach number
u = flow speed relative to stationary shock
U = shock speed relative to shock tube
a = speed of sound
P = pressure
γ = specific heat capacity
T = temperature
ρ = density
R = Boltzmann constant
Mol = molar mass.

The subscript in each property defines the property at that state. The initial pressure (P_1) and temperature (T_1) were measured from the pressure transducers. Mach number M_1 is calculated by considering the shock at the stationary position and the flow is moving into the shock. As the flow in front of the shock was in rest, such that the flow speed is $u_1 = U_I$, which gives,

$$M_1 = \frac{u_1}{\mathbf{a}_1}. \tag{4.14}$$

$$\frac{P_2}{P_1} = \frac{2\gamma M_1^2 - (\gamma - 1)}{(\gamma + 1)}. \tag{4.15}$$

$$\frac{T_2}{T_1} = \frac{[2\gamma M_1^2 - (\gamma - 1)][(\gamma - 1)M_1^2 + 2]}{(\gamma + 1)^2 + M_1^2}. \tag{4.16}$$

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma + 1) + M_1^2}{(\gamma - 1)M_1^2 + 2}. \tag{4.17}$$

$$M_2 = \sqrt{\frac{(\gamma - 1)M_1^2 + 2}{2\gamma M_1^2 - (\gamma - 1)}}. \tag{4.18}$$

$$\mathbf{a}_2 = \sqrt{\frac{\gamma \cdot R \cdot T_2}{\mathrm{Mol}}}. \tag{4.19}$$

$$u_2 = M_2 \mathbf{a}_2. \tag{4.20}$$

When the incident shock is reflected, the conditions behind the initial shock wave are now the initial conditions for the reflected shock wave and flow speed,

$$u_3 = u_2 + U_4. \tag{4.21}$$

For estimating properties behind the normal shock, the above relations can be used. However, Mach number ahead of the oblique shock, M_3 is calculated as,

$$M_3 = \frac{u_3 \mathrm{Sin}\beta}{\mathbf{a}_2}. \tag{4.22}$$

For a detail study of shock relations and their implementation, please refer to Mark (1958); Liepmann and Roshko (2001); Law (2010).

5 Results and Discussion

In this chapter, some results from the developed algorithms, which use one or multiple methods described in the previous Chapter 4 are presented and briefly discussed. The results and discussion are presented according to the timeline of the articles published or prepared in which the algorithms were introduced. The list of the articles published during this PhD is given below. Please note that, the front tracking part of Article IV was finished before Article III, however the application part took longer time than expected, so Article III was published before IV. Article V is well prepared at this point of time and will be submitted soon.

Article I: Open Active Contour Model For Front Tracking Of Detonation Waves. *Maharjan, S., Gaathaug, A.V., Lysaker, O.M.*: In: Proceedings of the 58th Conference on Simulation and Modelling, pp. 174–179. Linköping University Electronic Press, Sweden (2017)

Article II: An Image Processing Framework for Automatic Tracking of Wave Fronts and Estimation of Wave Front Velocity for a Gas Experiment. *Maharjan, S., Bjerketvedt, D., Lysaker, O.M.*: In: Representation, analysis and recognition of shape and motion FroM Image data, CCIS. 187 . Springer, (2018)

Article III: Wave Front Tracking in High Speed Videos Using a Dynamic Template Matching. *Maharjan, S.*: In: 9th Iberian Conference on Pattern Recognition and Image Analysis, LNCS . Springer, (2019)

Article IV: Processing of High Speed Videos of Shock Wave Boundary Layer Interactions. *Maharjan, S., Bjerketvedt, D., Lysaker, O.M.*: Production Process: Signal, Image and Video Processing, Signal (2020)

Article V: Information Extraction from High Speed Videos of Reflected Shock Wave Interaction With Boundary Layer.*Maharjan, S., Bjerketvedt, D., Lysaker, O.M.*, well prepared and will submitt to Shock Waves.

5.1 Article I: Open active contour model

Open Active Contour Model For Front Tracking Of Detonation Waves.

An open active contour algorithm was developed to track wave fronts in the high speed videos of a gas detonation experiments, one example is shown in Figure 3.5. The developed algorithm can be used to track the open edge of any object which expands from one end to another end in the image. The novelty of this work is an open contour model with a simple fixed boundary condition.

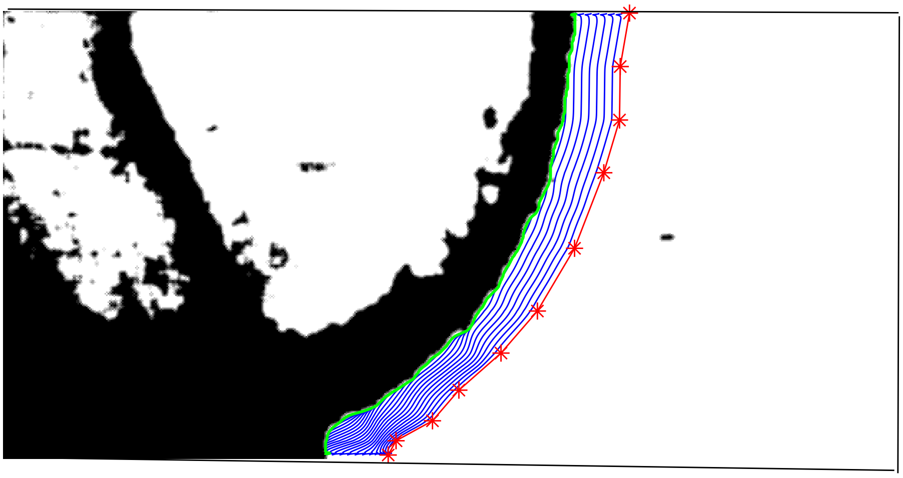


Figure 5.1: The progression of an initial snake (red) towards the wave front, the green curve: final front.

Figure 5.1 shows the progression of an initial snake (red line) to the final front (green curve) by an open active contour algorithm. The snake was initialized by few points shown as red asterisks in Figure 5.1. Even after the consideration of internal energy for the boundary points and tuning the α and β parameter, a close look at the boundaries shows slight error. However, as the number of boundary points are few with compare to the remaining points, these points can be excluded while estimating wave properties.

The drawback, however is similar to the classic snake models, the snake fails to move towards an object when the initial points are far from the object. An example is demonstrated in Figure 5.2, the snake didn't move around the lower boundary as the snake is too far from the front. As a result, the snake in the middle also get drawn to the local minima (pointed by an arrow) and could not come out as a result of internal force balance between lower and upper snake points. These drawbacks can be minimized by tuning the

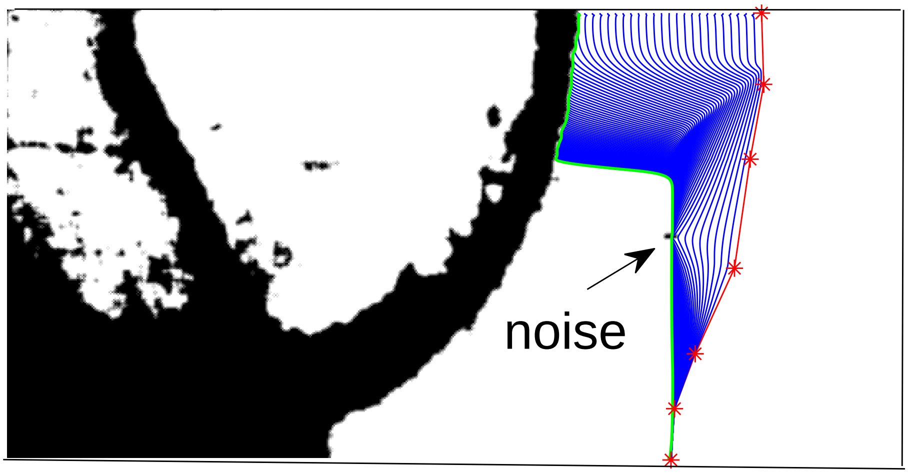


Figure 5.2: A progression of initial snake (red) towards the wave front, when the initial snake is far from the object, the green curve: final front.

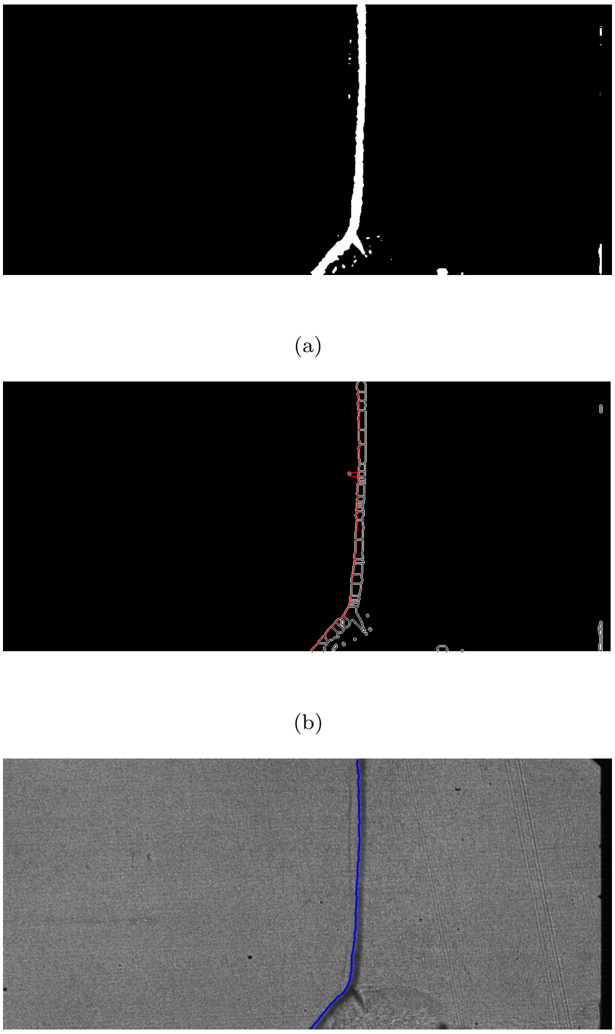
parameters or appropriately initializing the snake. While the initialization of the contour can be handled by using some additional information or using the previous contour, tuning parameters takes a lot of time. This problem became a huge challenge when the images within a high speed video are of different intensity levels and the algorithm has to be used in multiple videos.

5.2 Article II: Watershed algorithm

An Image Processing Framework for Automatic Tracking of Wave Fronts and Estimation of Wave Front Velocity for a Gas Experiment

An algorithm combining watershed segmentation and image morphology was developed to track the wave front in a high speed video of a shock wave (Figure 1.8). The advantage of the watershed algorithm over the active contour model is, it works comparatively faster, besides there was no need for initialization and any other parameters. The stepwise algorithm is described in Section 4.3, the result after some of the steps is presented in Figure 5.3.

Figure 5.3(a) is the result after image segmentation by applying Otsu's method and the corresponding morphologically removed image is in Figure 5.3(b). A morphological operation 'remove' sets a pixel to 0 if all its 4-connected neighbours are 1, thus leaving only the boundary pixels on (Gonzalez and Woods, 2000), which helps in removing noise, for example in Figure 5.3(b) behind the oblique shock, the number of white bulbs structure is less than in Figure 5.3(a).


(a)

(b)

(c)

Figure 5.3: Watershed algorithm in a high speed image (a) segmented image; (b) morphologically 'remove' image with initial front tracked as a first white pixel from left; (c) a final contour after post-processing on the corresponding raw image.

By using apriori information of the wave propagation direction, a front position was picked at the first pixel where the intensity value changes from 0 to 1, shown as a red curve in Figure 5.3(b). However, for a few images there were some false tracking due to oversegmentation (can be seen around the mid part in Figure 5.3(b)). As the most of the false tracking were around the normal shock, a post-processing by using a median of the normal front was used to overcome this problem. The final front is plotted as a blue curve upon a corresponding raw image in Figure 5.3(c).

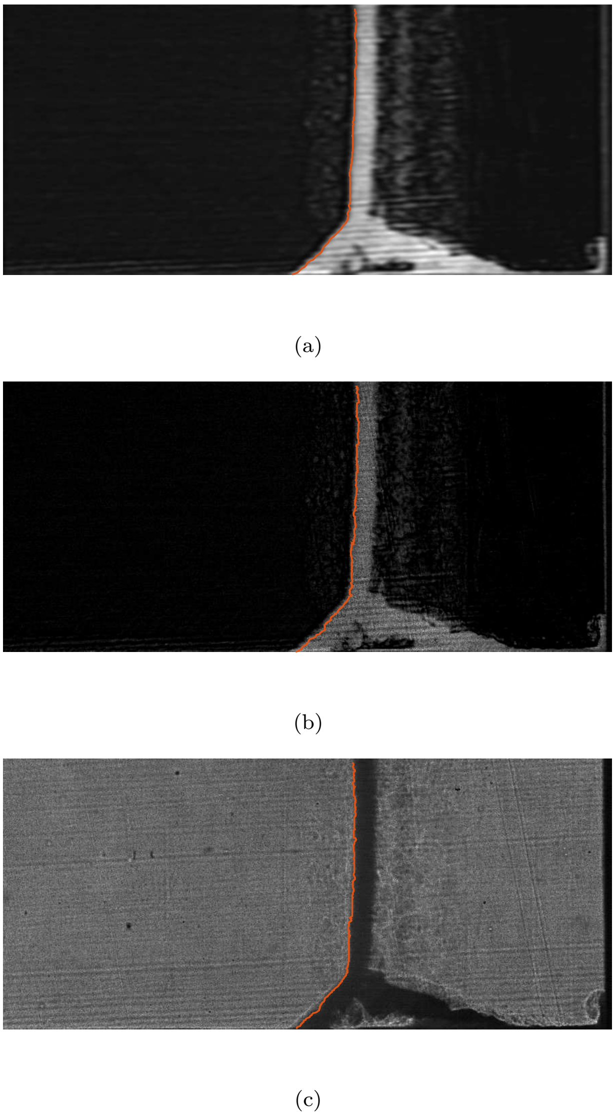
Most of the segmentation using the watershed algorithm used markers to overcome the problem of over segmentation. While developing watershed segmentation particularly for the problem in hand, over segmentation was encountered fairly low. As shown in the mid part in Figure 5.3(b)), there were small discontinuities that were corrected by post-processing. Besides, most of the over segmentation was behind the wave front which didn't have any impact. For this reason, this work doesn't dwell into using marker based watershed segmentation. However, using marker based segmentation could have been the next step in case of taking this algorithm further.

5.3 Article III: Dynamic template matching

Wave Front Tracking in High Speed Videos Using a Dynamic Template Matching

A dynamic template matching algorithm based on MSE was developed to track the wave fronts in the different grayscale images of a high speed video. A template whose value varies in each image depending on the maximum and minimum intensity was introduced. As can be seen in all the images presented in this thesis, the wave occupy a small part of the image with respect to the columns. Therefore, to minimize the processing time, a bounding box was created around the area of interest (wave) such that the sliding of the template occurs only inside the bounding box. With prior information about the size and direction of the wave in a high speed video, the size and initial position of the bounding box was manually set.

Figure 5.4(a) shows the result of the algorithm when implemented in a frequency filtered image. Similarly, the result of the same algorithm performed in background subtracted images can be seen in Figure 5.4(b). For tracking fronts in a raw image, the side of the template was swapped as the intensity of the front and background was opposite from the filtered images. The result in Figure 5.4(c) shows that the algorithm works almost as good as in filtered images.



(a)

(b)

(c)

Figure 5.4: Result of a dynamic template matching in (a) filtered image; (b) background subtracted image; (c) raw image.

Most of the previous work in the field of front tracking is based on edge detection techniques which include image threshold Cui et al. (2013). However, image thresholding especially in the low contrast images mostly results in an error of few pixels around the actual edges (Villarrubia et al., 2002). Some of the pixels around the front which might be part of the wave were segmented as a background. For ordinary problems, a false segmentation of few pixels might not be crucial. However, for this PhD, the positions of the fronts play an extremely crucial role as they are going to be used for speed calculation. The purposed method of dynamic template matching may be one of the simple and effective ways to eliminate image thresholding.

5.4 Article IV: Segmented regression

Processing of High Speed Videos of Shock Wave Boundary Layer Interactions

In this article, a binary template matching for front tracking and a segmented regression method for separating tracked wave fronts were developed. Also, we used a different filtering method i.e. low pass filter in the Fourier domain (refer to section 2.1.1 for frequency filter). A binary template (see Figure 4.5) was created to track the front position in the segmented images like in 5.3(a). The motivation behind this work was to eliminate all the morphological steps in Article II while minimizing false tracking.

A segmented regression method was introduced to estimate the optimum breaking point (BP_o) to divide a tracked front into a normal and an oblique shock front. Even after calculating the BP_o, to estimate a precise triple point, two lines were fitted again as shown in Figure 5.5; one with 100 points above the BP_o(blue) and one with all the below points (red). Now, the crossing point of these two lines shown as a white point is the position of a triple point and the slope of the red line then gives the shock angle.

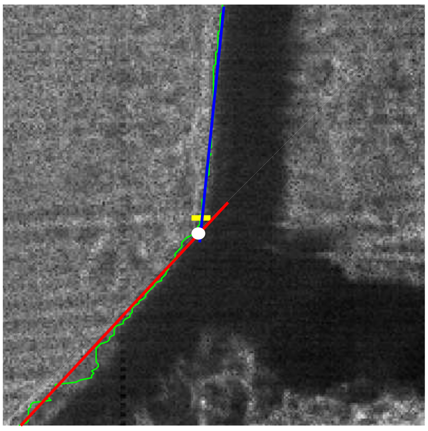


Figure 5.5: Segmented regression for estimating a triple point, yellow mark is BP_o and white mark is the triple point.

The limitation of this method was observed among few initial reflected shock, where the oblique shock is small, the line fitting for the oblique shock does show some error. Due to this, the first few fronts were excluded from this process. The estimated triple point in the series of fronts by using segmented regression along with the tracked fronts is shown in Figure 5.6.

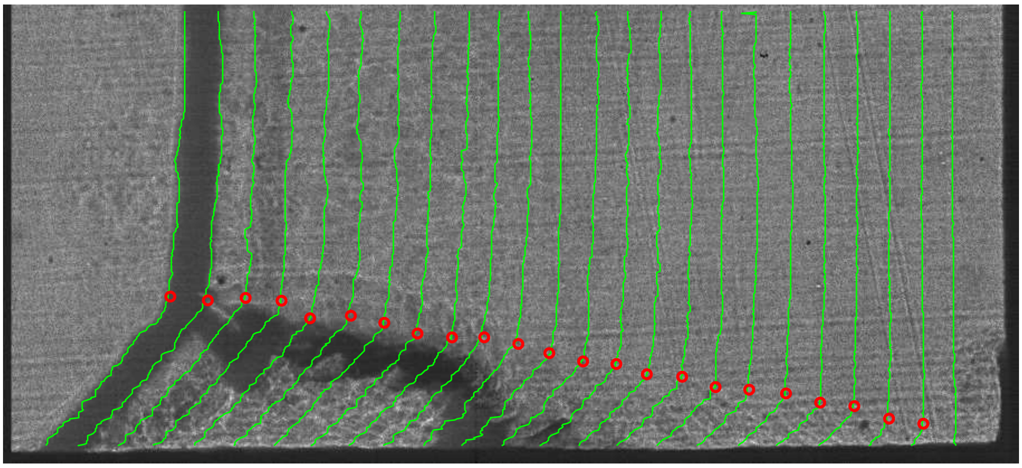


Figure 5.6: The triple point (red dots) determined by using a segmented regression method in one of the high speed video (Exp no. 2558). For better visual, only each 5^{th} front are plotted.

5.5 Article V: Shock wave information

Information Extraction from High Speed Videos of Reflected Shock Wave Interaction With Boundary Layer

Article V presents a method of estimating the shock wave information like shock speed, triple point, shock angles and the Mach number from the tracked wave fronts. The shock speed was calculated by using a NLSFM discussed in Section 4.6. This method helps to minimize the problem of oscillation drastically. Figure 5.7 shows the difference between shock speed calculated by using a two point method and by using NLSFM.

As each front contains at least 350-400 points, the segmented regression method was time consuming. A geometric approach using the previous triple point to find the current triple point was implemented as illustrated in Figure 5.8. If 'TP(1)' is a triple point a synthetic front 'Front 1' then the shortest distance between 'TP(1)' and 'Front 2' should be at 'TP(2)'. However, the actual tracked front is not exactly as straight as the synthetic ones. So, the median of the front is used to set the normal shock straight. Shock angles were estimated by fitting a straight line to the points in 'Front 2' that are below the calculated 'TP(2)'. The first triple point was manually set in the image/front in which it was visible.

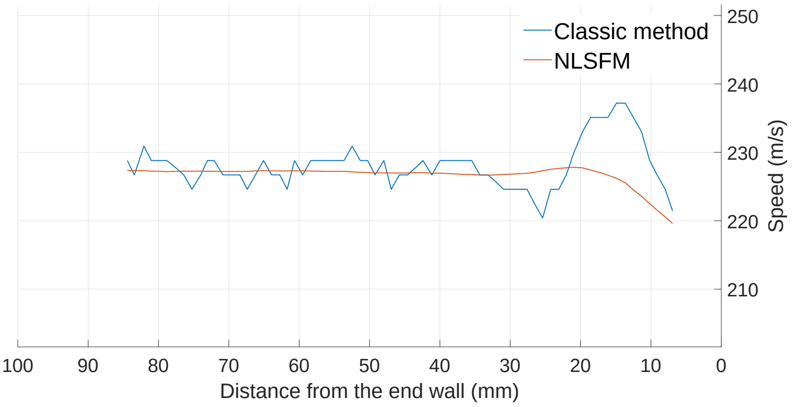


Figure 5.7: Oblique shock speed calculated by two point method (blue curve) and NLSFM (orange) (Exp no. 2519) with n = 10.

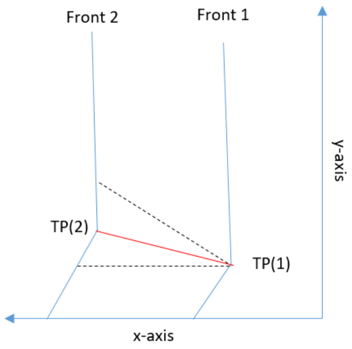


Figure 5.8: An illustration of determining current triple point by using previous triple point.

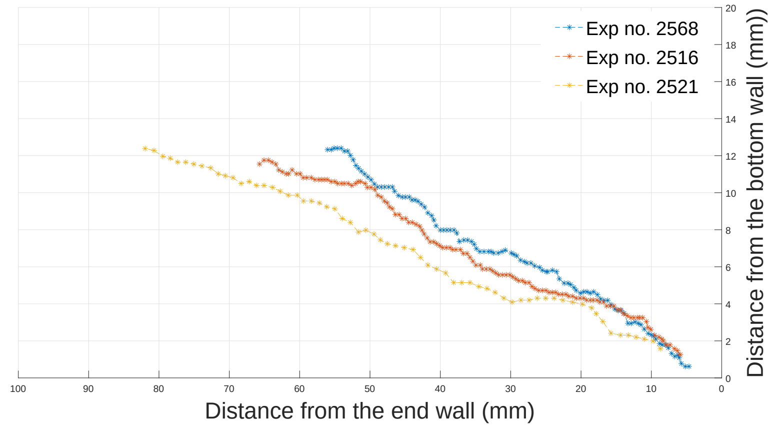


Figure 5.9: The calculated height of triple point from the lower boundary.

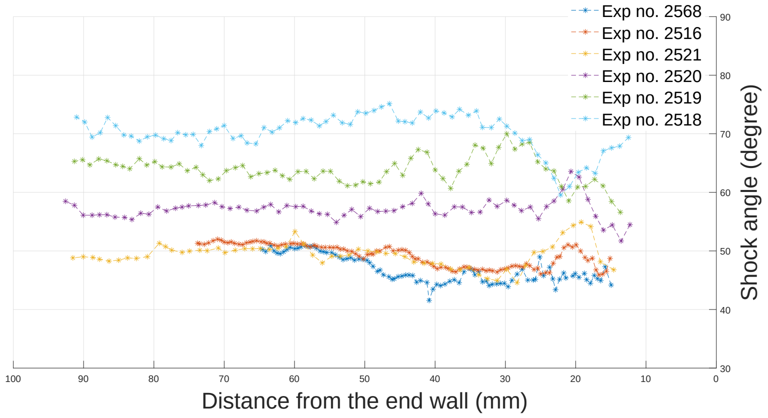


Figure 5.10: Shock angles calculated for all the experiments.

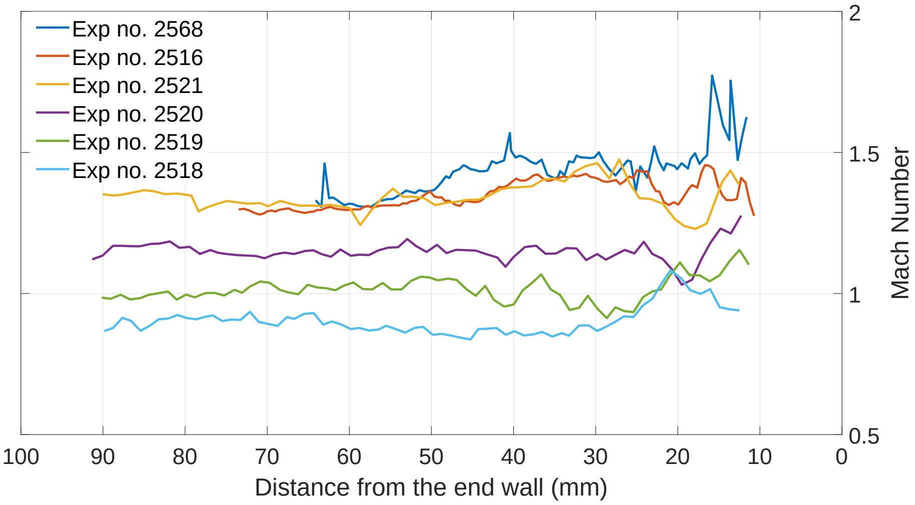


Figure 5.11: Mach number of the flow behind the oblique shock.

Six SWBLI experiments conducted with various initial pressure ranging from 7.5 - 20 kPa (an increase of 2.5 kPa) were processed in this work. The experiments are listed as a legend in the figures of this section, following top to bottom approach i.e. Exp. no. 2568 corresponds to the lowest pressure 7.5 kPa. The progression of triple points in three SWBLI experiments, where it exists are illustrated in Figure 5.9. The triple point structure vanishes during higher initial pressure experiments. Similarly, the estimated shock angle and Mach number behind the oblique shock M_4 are presented in Figure 5.10 and Figure 5.11 respectively. The Mach number M_4 was estimated by replacing M_1 and M_2 in (4.18) by M_3 (4.22) and M_4 respectively.

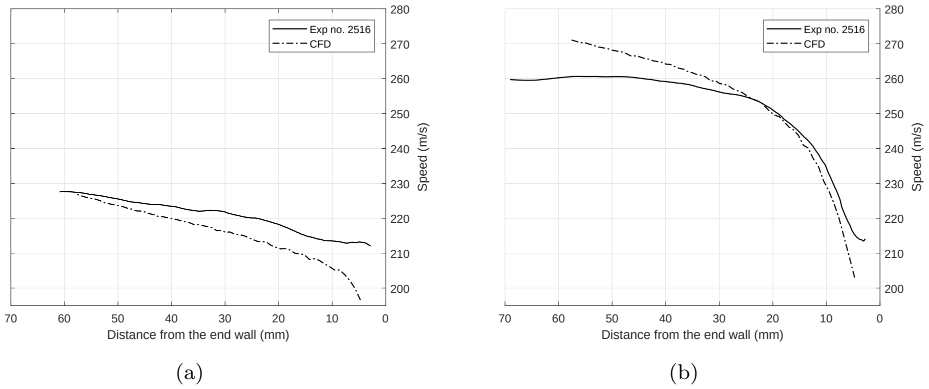


Figure 5.12: Comparision of estimated speed with CFD simulation (a) normal shock speed; (b) oblique shock speed (Exp no. 2516).

Comparision with CFD :

The estimated speed for SWBLI experiment (Exp no. 2516) was compared with the result from CFD simulation which are presented in Figure 5.12. It can be observed that, the results matches with each other. Due to the ideal condition used during CFD simulation, the results are smoother than the results from image processing.

This comparison is not included in the paper.

6 Conclusions

In this chapter, main conclusions of this thesis are discussed.

This PhD work presents the study of various image processing techniques conducted to process the high speed videos recorded during two different types of gas experiments. Gas detonation experiments were conducted at USN, Norway and SWBLI experiments were conducted at Caltech, USA. The main focus was to extract the information about the propagating wave, like the speed and the angles from the recorded high speed videos. These information were extracted by tracking/contouring a wave front in each image of the high speed videos. However, before implementing a tracking algorithm, images were first denoised by using a non-linear filter or a convolutional filter.

The first tracking algorithm was developed based on the active contour model to track the wave fronts in gas detonation experiments. The algorithm performs satisfactory, but the parameters need to be tuned properly, which is slightly time consuming. In addition, if the initial curve is far from the object to track, the convergence is slow and in some cases it doesn't converge at all. To eliminate the need for initializing, a second algorithm based on image segmentation was introduced. The algorithm consists of multiple steps and the position of the front needs to be determined after the image is segmented into a binary image. The basic approach of determining the position of the wave based on a transition of pixel value from 0 to 1 gives some misplacement with respect to the front location.

The third algorithm was based on a binary template matching and implemented to track a front in a binary image. This algorithm works comparatively faster and reduces mistracking observed sometime during the second algorithm. However, it was realized that the image thresholding itself was not favorable while estimating the shock speed based on the tracked position. An error while thresholding, even it is just 1 or 2 pixels, has a huge impact on shock speed estimation, due to the high frequency of the camera and a high time resolution. Therefore, the fourth algorithm based on dynamic template matching was developed to track the fronts in the grayscale images. In addition, in the framework of the fourth algorithm a bounding box around the wave position reduces the processing time. Each developed algorithm has its pros and cons, however, among the developed algorithms, dynamic template matching is the simplest and the fastest.

All the developed algorithms make use of a priori knowledge about direction of wave propagation to track the fronts. Moreover, the algorithms works step wise, first filtering, then tracking and in the end extracting information. Machine learning could be one way to eliminate this multi step process, an algorithm based on machine learning was partially developed. Even though it is not clear to which extent machine learning algorithm is a suitable framework for this particular problem, this can definitely be an exciting future work.

Regardless of a tracking algorithm, once the position of the wave fronts were located, the primary information; shock speed (both normal and oblique), position of a triple point and shock angle were calculated. Thereafter, by combining the calculated primary information and the traditional gas equations, secondary information like the Mach number, pressure, and temperature can be estimated. A basic approach of calculating speed i.e distance/time gave some oscillations in the shock speed even in cases when it is known to be constant. NLSFM was implemented instead which minimize the oscillations significantly.

The approach of studying gas dynamics, especially the SWBLI of the reflected shock wave using high speed imaging and image processing proved to be very informative. This approach in combination with gas dynamics equations enables to extract crucial information about the wave. The calculated wave properties demonstrate a detail variation that occurred within a time interval of 300 microseconds within a distance of 100 mm. This information is difficult to extract by using traditional approach of pressure readings. It can be concluded that the high speed films in combination with image processing can enrich and to some extent give new and more detailed insight to gas dynamics. In addition, the results from these algorithms can also be used as a validation for the experimental results and computer simulation.

Bibliography

(2011). *Particle Filter Based on Color Feature with Contour Information Adaptively Integrated for Object Tracking*, volume 2.

Akbar, R. (1997). Mach reflection of gaseous detonations.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916.

Arnold, E., Al-Jarrah, O. Y., Dianati, M., Fallah, S., Oxtoby, D., and Mouzakitis, A. (2019). A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20:3782–3795.

Babinsky, H. and Harvey, J. (2011). *Shock Wave-Boundary-Layer Interactions*. Cambridge University Press, Cambridge, UK.

Balaji, S. and Karthikeyan, S. S. (2017). A survey on moving object tracking using image processing. *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pages 469–474.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Beucher, S. and Bilodeau, M. (1979). Use of watersheds in contour detection. *International workshop on Image Processing: Real-time edge and motion detection/estimation*, pages 17–21.

Beucher, S. and Meyer, F. (1993). The morphological approach to segmentation: The watershed transformation. *Mathematical Morphology in Image Processing*, 34:433–481.

Bieniecki, W. (2004). Oversegmentation avoidance in watershed-based algorithms for color images. pages 169–172.

Bieniek, A. and Moga, A. (2000). An efficient watershed algorithm based on connected components. *Pattern Recognition*, 33(6):907–916.

Biswas, S. and Qiao, L. (2007). A comprehensive statistical investigation of schlieren image velocimetry (siv) using high-velocity helium jet. *Exp Fluids*, 58(18).

Bjerketvedt, D., Gaathaug, A., Vaagsaether, K., and Thomas, G. (2015). Front tracking of ddt from ultra-high speed video films. *Proceedings of the 25th International Colloquium on the Dynamics of Explosions and Reactive Systems (ICDERS)*, (19).

Brossard, C., Monnier, J., Barricau, P., Vandernoot, F., and Le Sant, Y. (2009). Principles and applications of particle image velocimetry. *The Onera Journal Aerospace Lab*, 1:1–11.

Brunelli, R. (2009). Template matching techniques in computer vision: Theory and practice.

Caselles, V., Catte, F., Coll, T., and Dibos, F. (1993). A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–31.

Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic active contours. *International Journal of Computer Vision*, 22:61–79.

Chan, T., Esedoglu, S., Park, F., Yip, M., Paragios, N., Chen, Y., and Faugeras, O. (2006). Recent developments in total variation image restoration. *Mathematical Models of Computer Vision*, pages 17–31.

Chan, T. and Shen, J. (2005). *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods.* Society for Industrial and Applied Mathematics., Philadelphia, USA.

Chan, T. and Vese, L. (1999). An active contour model without edges. *Scale-Space Theories in Computer Vision*, pages 141–151.

Chantara, W., Mun, J. H., Shin, D., and Ho, Y. (2015). Object tracking using adaptive template matching. *IEIE Transactions on Smart Processing and Computing*, 4:1–9.

Chen, Y. and Pock, T. (2017). Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272.

Chenyang, X. and Jerry, L. P. (1997). Gradient vector flow: a new external force for snakes. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 66–71.

Ciaparrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., and Herrera, F. (2020). Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88.

Cohen, L. (1991). On active contour models and balloons. *CVGIP: Image Understanding* ., 53(2):211–218.

Cohen, L. and Cohen, I. (1993). Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147.

Cohen, L.and Kimmel, R. (1996). Global minimum for active contour models: A minimal path approach. volume 24, pages 666–673.

Cui, S., Wang, Y., Qian, X., and Deng, Z. (2013). Image processing techniques in shockwave detection and modeling. *Journal of Signal and Information Processing*, 04:109–113.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095.

Damazo, J. (2013). *Planar Reflection of Gaseous Detonations*. Doctoral thesis, California Institute of Technology, Pasadena, California.

Della Rocca, M., Fiani, M., Fortunato, A., and Pistillo, P. (2004). Active contour model to detect linear features in satellite images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34.

Dokladal, P., Enficiaud, R., and Dokladalova, E. (2004). Contour-based object tracking with gradient-based contour attraction field. volume 3, pages iii – 17.

Duda, R. and Hart, P. (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15.

Ebbecke, M., Ali, M. B. H., and Dengel, A. (1997). Real time object detection, tracking and classification in monocular image sequences of road traffic scenes. 2:402–405 vol.2.

Estruch, D., Lawson, N. J., MacManus, D. G., Garry, K. P., and Stollery, J. L. (2008). Measurement of shock wave unsteadiness using a high-speed schlieren system and digital image processing. *Review of Scientific Instruments*, 79(12).

Farabet, C., Couprie, C., Najman, L., and Lecun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35:1915–1929.

Fisher, R., Perkins, S., Walker, A., and Wolfart, E. (2003). *Spatial Filters - Laplacian of Gaussian*.

Gaathaug, A. V., Vaagsaether, K., Knudsen, V., and Bjerketvedt, D. (2012). Experimental and numerical investigation of ddt in hydrogen–air behind a single obstacle. *International Journal of Hydrogen Energy*, 37(22):17606–17615.

Ganin, Y. and Lempitsky, V. (2015). Neural network nearest neighbor fields for image transforms. pages 536–551.

Gibou, F., Fedkiw, R., and Osher, S. (2017). A review of level-set methods and some recent applications. *Journal of Computational Physics*, 353:82–109.

Girshick, R., Donahue, J.and Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 25:580–587.

Glass, I. and Patterson, G. (1955). A theoretical and experimental study of the shock tube. *Journal of the Aeronautical Sciences*, 23(22):73–100.

Gonzalez, C. and Woods, E. (2000). *Digital Image Processing.* Prentice-Hall, New Jersey, USA.

Guoshuai, L., Muhammed, B. A., Konstantinos, K., Takahiro, U., and Sriram, R. (2019). Image processing techniques for shock wave detection and tracking in high speed schlieren and shadowgraph systems. *Journal of Physics: Conference Series*, 1215:012021.

Haglund, J. T. J. (2015). Detonation in stratified hydrogen layers.

Hai, T., Sawhney, H. S., and Kumar, R. (2002). Object tracking with bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):75–89.

Hardie, R. and Barner, K. (1994). Rank conditioned rank selection filters for signal restoration. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 3:192–206.

Hoheisel, M. (2006). Review of medical imaging with emphasis on x-ray detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 563:215–224.

Hongsheng, L., Shen, T., Smith, M. B., Fujiwara, I., Vavylonis, D., and Huang, X. (2009). Automated actin filament segmentation, tracking and tip elongation measurements based on open active contour models. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1302–1305.

Hough, P. (1962). Method and means for recognizing complex patterns.

Hu, W., Li, W., Zhang, X., and Maybank, S. (2015). Single and multiple object tracking using a multi-feature joint sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):816–833.

Jain, V. and Sebastian, S. (2009). Natural image denoising with convolutional networks. pages 769–776.

Kaiheng, D., Yuehuan, W., and Qiong, S. (2019). Real-time object tracking with template tracking and foreground detection network. *Sensors (Basel, Switzerland)*, 19.

Kanagamalliga, S. and Vasuki, S. (2018). Contour-based object tracking in video scenes through optical flow and gabor features. *Optik*, 157:787–797.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kilger, M. (1992). A shadow handler in a video-based real-time traffic monitoring system. pages 11–18.

Kimmel, R., Kiryati, N., and Bruckstein, A. M. (1996). Sub-pixel distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision*, 6(2):223–233.

Kivinen, J., Williams, C., and Heess, N. (2014). Visual boundary prediction: A deep neural prediction network and quality dissection. volume 33, pages 512–521.

Kleine, H., Lyakhov, V. N., Gvozdeva, L. G., and Gronig, H. (1992). Bifurcation of a reflected shock wave in a shock tube. pages 261–266.

Kleine, H., H. K. M. H. H. T. Y. J. K. K. K. Y. E. T. G. (2005). High-speed time-resolved color schlieren visualization of shock wave phenomena. *Shock Waves*, 14:333–341.

Konishi, S., Yuille, A. L., Coughlan, J. M., and Zhu, S. C. (2003). Statistical edge detection: learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):57–74.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25:1106–1114.

Law, C. (2010). *Combustion Physics*. Cambridge University Press, New York, USA.

Lee, J., Hua, F., and Jang, J. (2009). An improved object detection and contour tracking algorithm based on local curvature. In Ślęzak D., Pal S.K., K. B. G. J. K. H. K. T. e., editor, *Signal Processing, Image Processing and Pattern Recognition. SIP 2009. Communications in Computer and Information Science*, volume 61, pages 25–32. Springer, Berlin, Heidelberg.

Li, J. (2003). *A Wavelet Approach to Edge Detection*. Doctoral thesis, Sam Houston State University, Texas, USA.

Liepmann, H. and Roshko, A. (2001). *Elemnents of Gasdynamics*. Dover Publications Inc, New York, USA.

Lim, J. S. (1990). *Two-dimensional Signal and Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Lysaker, M., Lundervold, A., and Tai, X.-C. (2003). Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Transactions on Image Processing*, 12(12):1579–1590.

Mahalakshmi, T., Muthaiah, R., and Swaminathan, P. (2012). An overview of template matching technique in image processing. *Research Journal of Applied Sciences, Engineering and Technology*, 4(24):5469–5473,.

Maier, A., Syben, C., Lasser, T., and Riess, C. (2018). A gentle introduction to deep learning in medical image processing. *Zeitschrift für Medizinische Physik*, 29(2):86–101.

Mark, H. (1958). *The Interaction of a Reflected Shock Wave With the Boundary layer in a shock tube*. Doctoral thesis, Cornell University, Ithaca, New York.

Marr, D. and Hildreth, E. C. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, 207:187–217.

Martin, D. R., Fowlkes, C. C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549.

Mata, H. O., Falcapoundso Filho, J. B. P., Avelar, A. C., Carvalho, L. M. M. d. O., and Azevedo, J. L. F. (2017). Visual experimental and numerical investigations around the vlm-1 microsatellite launch vehicle at transonic regime. *Journal of Aerospace Technology and Management*, 9:179 – 192.

Mckenna, S., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H. (2000). Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56.

Meenakshi, S. and Gomathy, N. (2015). A survey on real time object detection and tracking algorithms. *International Journal of Applied Engineering Research*, 10:8290–8297.

Melonakos, J., Pichon, E., Angenent, S., and Tannenbaum, A. (2008). Finsler active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):412–423.

Meyer, F. (1994). Meyer, f.: Topographic distance and watershed lines. signal process. 38, 113-125. *Signal Processing*, 38:113–125.

Meyer, F. and Beucher, S. (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation - JVCIR*, 1:21–46.

Mumford, D. and Shah, J. (1989). Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685.

Murray, D. and Basu, A. (1994). Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):449–459.

Najim, M. (2010). *Digital filters design for signal and image processing.* Wiley Press, London, UK.

NASA (1999). Making new reality: Computers in simulations and image processing.

Ojha, S. and Sakhare, S. (2015). Image processing techniques for object tracking in video surveillance- a survey. pages 1–6.

Osher, S. and Sethian, J. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Ouyang, W., Tombari, F., Mattoccia, S., Stefano, L., and Cham, W. (2012). Performance evaluation of full search equivalent pattern matching algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):127–143.

Patel, C. and Patel, R. (2012). Contour based object tracking. *International Journal of Computer and Electrical Engineering.*

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Piccardi, M. (2004). Background subtraction techniques: a review. 4:3099–3104 vol.4.

Pohl, W. G. (2002). Peter salcher und ernst mach schlierenfotografie von ‹berschall-projektilen, wie diese bilder unsere sinne erweiterten und in welcher weise sie ein beispiel f¸r wissenschaftliche zusammenarbeit im habsburgerreich sind. pages 22–26.

Prewitt, J. (1979). *Object enhancement and extraction.* Academic Press.

Raffel, M. (2015). Background-oriented schlieren (bos) techniques. *Experiments in Fluids*, 56(60).

Ranzato, M., Boureau, Y., Chopra, S., and Lecun, Y. (2007). A unified energy-based framework for unsupervised learning. *Proc. Eleventh International Conference on Artificial Intelligence and Statistics.*

Roberts, L. (1963). *Machine perception of three-dimensional solids.* Doctoral thesis, Massachusetts Institute of Technology, Massachusetts, USA.

Romero-Záliz, R. and Reinoso-Gordo, J. F. (2018). An updated review on watershed algorithms.

Rosenfeld, A. (1969). Picture processing by computer. *ACM Comput. Surv.*, 1(3):147–176.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268.

Ryan, S. and Laurie, S. (2007). A tutorial on the piecewise regression approach applied to bedload transport data.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Senthilkumaran, N. and Reghunadhan, R. (2007). Edge detection techniques for image segmentation - a survey of soft computing approaches. *INFORMATION PAPER International Journal of Recent Trends in Engineering*, 1.

Settle, G. (2001). *Schlieren and Shadowgraph Techniques*. Springer, Berlin, Heidelberg, Berlin, Heidelberg.

Settle, G. and Hargather, M. (2017). A review of recent developments in schlieren and shadowgraph techniques. *Measurement Science and Technology*, 28(4).

Shemesh, M. and Ben-Shahar, O. (2011). Free boundary conditions active contours with applications for vision. In *Bebis G. et al. (eds) Advances in Visual Computing. ISVC 2011. Lecture Notes in Computer Science*, volume 6938, Berlin, Heidelberg. Springer, Berlin, Heidelberg.

Shimamura, P. (2015). Muybridge in motion: Travels in art, psychology and neurology. *History of Photography*, 26:341–350.

Sobel, I. (2014). An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*.

Sommersel, O. K., Bjerketvedt, D., Christensen, S. O., Krest, O., and Vaagsaether, K. (2008). Application of background oriented schlieren for quantitative measurements of shock waves from explosions. *Shock Waves*, 18(4):291–297.

Song, E. Y. and Lee, J.-J. (2004). Moving object detection using region tracking. *Artificial Life and Robotics*, 8:20–28.

Stephen, G., Gao, T., and Koller, D. (2009). Region-based segmentation and object detection. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 655–663. Curran Associates, Inc.

Supreeth, H. S. G. and Patil, C. M. (2018). Moving object detection and tracking using deep learning neural network and correlation filter. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1775–1780.

Timmerman, B. (2008). High-speed digital visualization and high-frequency automated shock tracking in supersonic flows. *Optical Engineering*, 47:103201.

Tom, G. and Osher, S. (2009). The split bregman method for l1-regularized problems. *SIAM J. Imaging Sciences*, 2:323–343.

Vaagsaether, K. (2010). *Hydrogen Leaks in Partially Confined spaces-Dispersion and Explosions*. Doctoral thesis, University of South Eastern Norway/NTNU, Porsgrunn, Norway.

Vaagsaether, K., Knudsen, V., and Bjerketvedt, D. (2007). Simulation of flame acceleration and ddt in h2–air mixture with a flux limiter centered method. *International Journal of Hydrogen Energy*, 32(13):2186–2191.

van Dijk, N. P.and Maute, K., Langelaar, M., and van Keulen, F. (2013). Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472.

Villarrubia, J., Vladár, A., Lowney, J., and Postek, M. (2002). Scanning electron microscope analog of scatterometry. *Proc. SPIE*, 4689.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:583–598.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. (2010). Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408.

Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: realtime tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.

Xizhi, Z. (2008). The application of wavelet transform in digital image processing. *2008 International Conference on MultiMedia and Information Technology*, pages 326–329.

Yamamoto, S., Tagawa, Y., and Kameda, M. (2015). Application of background-oriented schlieren (bos) technique to a laser-induced underwater shock wave. *Experiments in Fluids*, 56.

Yang, R., Yin, L., Gabbouj, M., Astola, J., and Neuvo, Y. (1995). Optimal weighted median filtering under structural constraints. *IEEE Transactions on Signal Processing*, 43(3):591–604.

Yilmaz, A., Xin, L., and Shah, M. (2004). Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536.

You, Y.-L. and Kaveh, M. (2000). Fourth-order partial differential equations for noise removal. *IEEE Transactions on Image Processing*, 9(10):1723–1730.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

Zhou, Y., Hu, B., and Zhang, J. (2006). Occlusion detection and tracking method based on bayesian decision theory. page 474–482. Springer-Verlag.

Part II

Published and unpublished scientific articles

Article I: Open Active Contour Model For Front Tracking Of Detonation Waves

Open Active Contour Model For Front Tracking Of Detonation Waves.
Samee Maharjan, Andre V. Gaathaug and Ola Marius Lysaker.
Published in: Proceedings of the 58th Conference on Simulation and Modelling, pp. 174–179. Linköping University Electronic Press, Sweden (2017)

OPEN ACTIVE CONTOUR MODEL FOR FRONT TRACKING OF DETONATION WAVES

Samee Maharjan [1] Andre Vagner Gaathaug[1] Ola Marius Lysaker[1]

[1]Department of process, energy and environmental technology, University College of Southeast Norway, Norway.
samee.maharjan@usn.no

Abstract

This paper presents an image processing framework for tracking the front of the detonation wave from a sequence of images. The images are captured by high speed camera during a laboratory gas explosion experiment. By tracking the fronts in two or three consecutive frames, it is possible to calculate the thermodynamic properties like velocity and pressure along the entire wave front. Alternatively, these calculations are limited to measurements recorded by sensors at some fixed, locations. An active contour model having Gradient Vector Flow (GVF) as an external force field is used to track the wave front in each image. The structure and the properties of detonations in combustion physics has been the point of interest since early 80's. In the present paper, detonation is studied in the stratified layer of combustible gas above a non-reacting layer of air. The recorded images are digitally processed, and the local velocities are calculated based on the tracked fronts. The calculated velocities are then used to estimate the pressure ahead of the wave front with the help of the normal shock relations. The estimated pressure is compared with the measured values from pressure transducers mounted on the top and bottom of the experiment tube.

Keywords: Open contour, Front Tracking, Detonation

1 Introduction

Active contours, popularly known as snakes, have been an essential part in image processing and computer vision applications. Snakes are mainly used for edge detection and boundary contouring in the field of image segmentation. A snake is a moving curve within an image, which eventually lie itself around the surface/edge of the desired object. The snake moves under the influence of internal forces within the curve itself and the external force calculated from the image data. It was first developed by Kass et al.in 1988 (Kass et al., 1988). Many developments and improvements have been purposed thereafter, for example the Chan-Vese (CV) model (Chan and L.Vese, 2001) that detects the boundary of an object using the Mumford-Shah functional (Mumford and Shah, 1989), the level set method (Osher and Sethian, 1988) to name a few. The application of active contours are seen mainly in the field of medical research see, (Phama and Tranc., 2015; Yan et al., 2014) and the references therein. For the last decade, active contour models have been an active research field. However most applications are based on closed contours. But for some application like front tracking, where the edge/boundary expands from top to bottom of the image an open snake is more efficient. This study will show the development of an open contour model and its application in the field of physical science of detonations.

Detonations are among the worst consequences of accidents related to gas handling and explosion in which a generated wave can exert pressure around 40 bar and velocity of more than 2000 m/s. It is relevant for many combustible gases, while mostly considered for the more reactive gases such as propane, ethylene, hydrogen and acetylene (Law, 2010). The detonation study of this work regards the detonations in a stratified layer of hydrogen gas above a non-reacting layer of air. The formation of a high-pressure Mach stem at the lower wall (see Figure 4) is of particular interest in the field of Combustion.

The paper is organized as follows: In Section 2, there is a short description of the experimental setup that was used for the experiment and image recording. In Section 3, a general snake model is described which is followed by Section 4, where the numerical implementation of an open snake is shown. The front tracking and velocity calculation is put together in Section 5. Finally, the results and the conclusions of the study is presented in Section 6.

2 Experimental Setup

Figure 1 illustrates the experimental setup for conduting the gas explosion experiments. The setup is made of a 3 m long channel with transparent polycarbonate walls. One end is closed and the other is open to the atmosphere, and an adjustable baffle type obstacle is located at 1 m from the closed end. The channel was filled with hydrogen and air and then ignited from the closed end such that the wave propagated towards the open end. Kistler 603b type pressure transducers were used to record the pressure at various locations along the top and the bottom wall of the tube. A high speed camera recorded the wave propagation behind the obstacle at 500 kHz. A schlieren imaging method was used to record the propagation at 23 cm diameter window expanding from a 1.2 m to 1.4 m behind the obstacle. The schlieren system is a double mirror system which is based on the fact that light rays

DOI: 10.3384/ecp17138174

bends whenever there is a change in the refractive index of the meduim in which the light passed (Settles, 2001). All experiments are done with premixed hydrogen and air at ambient temperature and pressure. Detail description of the setup can be found in (Gaathaug et al., 2010).

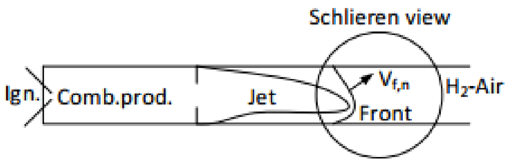


Figure 1. Experimental setup for premixed hydrogen and air combustion in a long tube.

Figure 2(a) shows the portion of captured high speed frame sequence during an experiment and Figure 2(b) shows the schlieren images captured at the highlighted rectangle portion in Figure 2(a). An overview of the overall designed framework for tracking the fronts from the image is depicted in the flowchart shown in Figure 3.

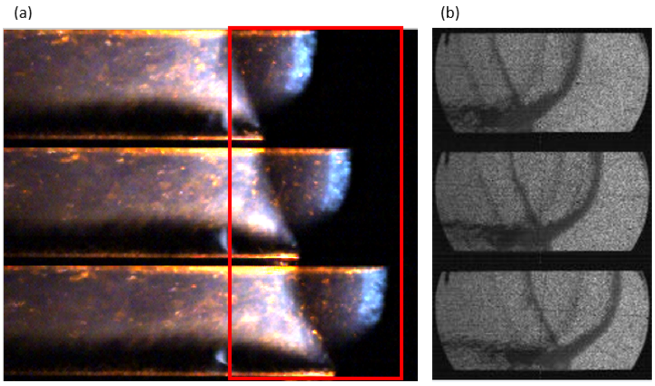


Figure 2. (a) High speed images showing a detonation wave in an experiment tube.(b) The schlieren images captured within the red rectangle in (a).

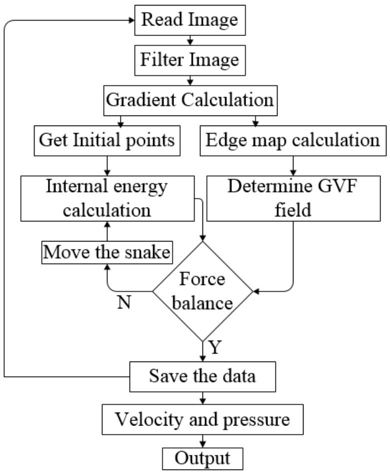


Figure 3. Flow chart showing the designed framework.

3 Active Contour Model

A basic snake defined in (Kass et al., 1988) is a parametric curve formed within an image. The snake is initialized manually by a set of *x*-*y* points around the object of interest and simultaneous interpolates between the points. By representing the position s of the snake parametrically as $V(s) = (x(s), y(s))$ in x-y coordinate system , its energy function is defined as in (1). The energy is calculated for each point along the snake and the snake will move in the direction where there is minimum energy compare to the previous position. The snake will be stationary once the energy is balanced in all directions.

$$E_{snake} = \int_s E_{int}(V(s)) + E_{ext}(V(s))ds. \tag{1}$$

The first term in (1), E_{int} is an internal energy which can be defined as

$$E_{int} = 1/2[\alpha(s)|V'(s)|^2 + \beta(s)|V''(s)|^2]. \tag{2}$$

Here $\alpha(s)$ and $\beta(s)$ are positive weighting parameters for controlling the snake's tension and rigidity respectively. $V'(s)$ and $V''(s)$ are the first and the second derivative of $V(s)$ with respect to s. The second term in (1), E_{ext} is the external energy function which is calculated from the image such that it takes minimum values at the point of interest like edges and boundaries. For example, for a gray scale image $Im(x, y)$, the external energy can be calculated as

$$E_{ext} = -|\nabla Im(x, y)|^2, \tag{3}$$

where ∇ is the gradient operator. A snake that minimizes E_{snake} must also satisfy the Euler equation

$$\alpha V''(s) - \beta V''''(s) - \nabla E_{ext} = 0, \tag{4}$$

which can be viewed as force balance between the internal and the external forces. Both $\alpha(s)$ and $\beta(s)$ are taken as constant for the entire framework. Mathematically,

$$F_{int} + F_{ext} = 0. \tag{5}$$

The internal force term $F_{int} = \alpha V''(s) - \beta V''''(s)$ prevents the snake from stretching and bending whereas the external force term $F_{ext} = -\nabla E_{ext}$ attracts the snake towards the desired location (Xu and Prince, 1997).
To solve (4), $V(s)$ is taken also as a function of time t such that $V(s)$ moves with time step t i.e. $V(s, t)$. The external energy E_{ext} for the image does not change with time.

$$\frac{\partial V(s,t)}{\partial t} = \alpha V''(s,t) - \beta V''''(s,t) - \nabla E_{ext}. \tag{6}$$

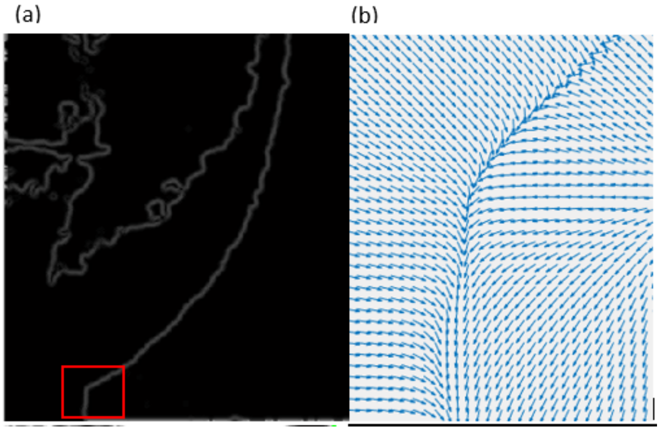


Figure 4. (a) Edge map of the image, Mach stem area is highlighted with red square. (b) External force field around the Mach stem.

4 Numerical Implementation

The numerical solution of (6) can be found when $\frac{\partial V(s,t)}{\partial t}$ tends to zero and (6) becomes,

$$\alpha V''(s,t) - \beta V''''(s,t) - \nabla E_{ext} = 0. \tag{7}$$

The solution for (7) can be obtained by decritizing the continous snake $V(s,t)$ as in (8). The details can be found in (Kass et al., 1988).

$$\begin{aligned}\alpha[V_i - V_{(i-1)}] - \alpha[V_{(i+1)} - V_i] + \beta[V_{(i-2)} - 2V_{(i-1)} + V_i]\\ -2\beta[V_{(i-1)} - 2V_i + V_{(i+1)}] + \beta[V_i - 2V_{(i+1)} + V_{(i+2)}]\\ +(f_x(i), f_y(i)) = 0.\end{aligned} \tag{8}$$

Here $V_i = (x_i, y_i)$ is the i^{th} point of the contour/snake s and $(f_x(i), f_y(i))$ which are equal to $(-\partial E_{ext}/\partial x_i$, $-\partial E_{ext}/\partial y_i)$ are the external force at point i.

4.1 Internal energy

The overall internal energy at each point can now be described in a matrix form as

$$AV + (f_x(i), f_y(i)) = 0. \tag{9}$$

Here A is pentadiagonal matrix ,

$$A = \begin{bmatrix} \beta & 0 & 0 & 0 & 0 \\ 0 & \alpha-3\,\beta & 0 & 0 & 0 \\ 0 & 0 & 2\,\alpha+6\,\beta & 0 & 0 \\ 0 & 0 & 0 & -\alpha-4\,\beta & 0 \\ 0 & 0 & 0 & 0 & \beta \end{bmatrix}$$

and V is a vector of the 5 consecutive points,
$V = [V_{i-2} \quad V_{i-1} \quad V_i \quad V_{i+1} \quad V_{i+2}]^T$.

If n is the total number of points in each contour, then for closed contour $V_1 = V_n$, the internal energy can be estimated using (8) for all points along the contour. However, for open contours the end points should remain at the top and bottom boundary, hence $V_{(i-1)}$ and $V_{(i-2)}$ will not exists for the first two points whereas $V_{(i+1)}$ and $V_{(i+2)}$ for the last two points. The matlab program is designed in such a way that these points stays at the boundary and therefore only move in x-direction with respect to its own external energy and its own previous position. Using Euler method with time step t for (8), the contour point V_i^t at time t can be related to its previous point V_i^{t-1} as,

$$AV_i^t + (f_x(i), f_y(i))V_i^{t-1} = -\gamma(V_i^t - V_i^{t-1}) \tag{10}$$

where γ is a step size. Solving (10) by matrix inversion and separating V_i into x_i and y_i,

$$x_i^t = (A + \gamma I)^{-1}[\gamma x_i^{(t-1)} - f_x(x_i^{(t-1)}, y_i^{(t-1)}] \tag{11}$$

$$y_i^t = (A + \gamma I)^{-1}[\gamma y_i^{(t-1)} - f_y(x_i^{(t-1)}, y_i^{(t-1)}], \tag{12}$$

where I is an identity matrix.

4.2 External energy

One of the main drawback of the snake models is that the calculated external energy were not sufficient for attracting the snake from long distance. The simple external energy calculated by using (3) works nicely when the initial snake is near to the object but fails to attract the snake when it is placed far away. To overcome this drawback, the gradient vector flow (GVF) proposed in (Xu and Prince, 1997) is used for the external force field. For estimating GVF, first the edge map $E_{edge}(x,y)$ is formed in such a way that it has a larger value at the edges and at the boundaries, compared to homogeneous region. Later, the estimated edge map is used to estimate the over all external force field. For this study, the edge map is taken as the gradient magnitude of the image,

$$E_{edge}(x,y) = \sqrt{(g_x^2 + g_y^2)}, \qquad [g_x, g_y] = \nabla Im(x,y).$$

and the result in shown in Figure 4(a). The Mach stem is marked by a red rectangle in this figure. Following the steps described (Xu and Prince, 1997), the final external external force $(f_x(i), f_y(i))$ for an image is calculated by iterating the equations (13) and (14) until it converges.

$$f_x(i) = \mu\nabla^2 f_x(i) - |\nabla E_{edge}(x)|^2(f_x(i) - \nabla E_{edge}(x)) \tag{13}$$

$$f_y(i) = \mu\nabla^2 f_y(i) - |\nabla E_{edge}(y)|^2(f_y(i) - \nabla E_{edge}(y)) \tag{14}$$

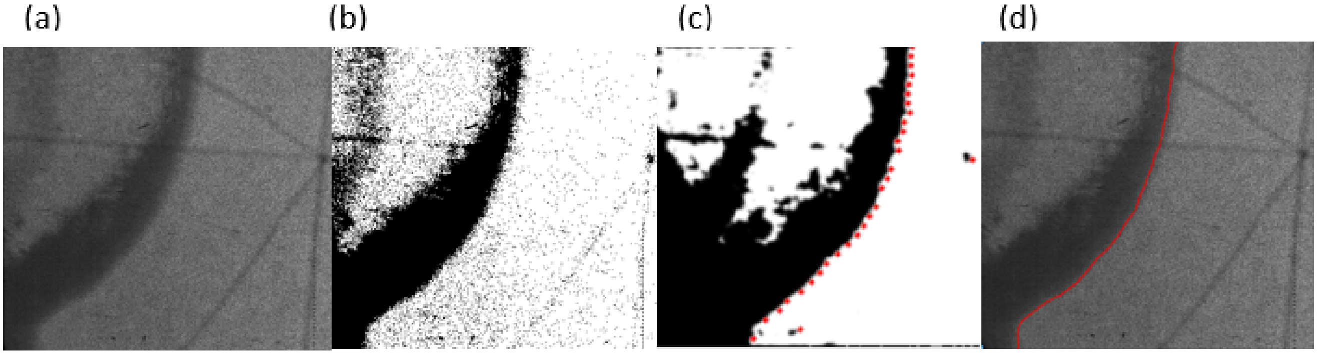


Figure 5. (a) Original raw image (b) Binary image (c) Filtered binary image with initial points for the snake. (d) Final contour along the wave front.

Here ∇^2 is the lapalcian operator and μ is a regularization parameter governing the trade off between the first term and second term. Generally, μ is taken as a constant which is set according to the noise present in the image. For a sequence of images where the noise varies with time, μ along with $|\nabla E_{edge}(x,y)|^2$ can be taken as spatially varying weighting factor as presented in (Xu and Prince, 1998). The initial value for $(f_x(i), f_y(i))$ is taken as $\nabla E_{edge}(x,y)$. The computed GVF field points towards the edges and varies smoothly over homogeneous regions, see Figure 4(b).

5 Front tracking and velocity calculation

It is necessary to reduce the background noise presents in the images before actual tracking of the front. For this purpose, the pre-processing of the image is done by first changing each image into binary form using Otsu method (Otsu, 1976), follwed by filtering using median filter. Figure 5(a) shows the raw image from the experiment and in Figure 5(b) the binary version of the same image is visualized. Though the GVF force field has advantages compared to the external forces used in (Kass et al., 1988), the formation of the initial points for the snake highly influence the overall performance of the framework. One possible way to initialize the snake automatically to a location close to the object, is to use the information of the gradient values in the image. For this task, a priori information of the direction of the wave propagation is used. By starting a search from the opposite direction of the wave propagation for predefined set of rows (every 20^{th}), an initial point is chosen to be the first point where the gradient value exceeds a given threshold value, see Figure 5(c). The snake is then initialized by interpolating these initial points such that there exist the snake point $V(s)$ for each unique row value of the image. For example, if size of an image Im is $[300 \times 800]$ then the size of the snake is $[300 \times 1]$. By implementating, the method defined in Section 4, the final contour is obtained as shown in Figure 5(d).

Suppose $V^K(s)$ is the final snake (tracked front) in K^{th} image of the sequence. For calculation of a normal velocity at point $V_i = (x_i, y_i)$ of the front $V^K(s)$, a local normal vector $\overrightarrow{n}$ is calculated by forming a local polynomial function around the point V_i (5 points above and 5 points below are used). The normal vector $\overrightarrow{n}$ will forms a triangle between the fronts with an angle θ at point V_i. The normal displacement is now estimated by using the final snake of the $V^{K+1}(s)$ of $(K+1)^{th}$ image of the sequence as shown in Figure 6. By using the value of $V^{K+1}(s)$ for the same row y_i the displacement d_m can be calculated and an angle θ can be estimated by calculating the slope of the normal vector $\overrightarrow{n}$.

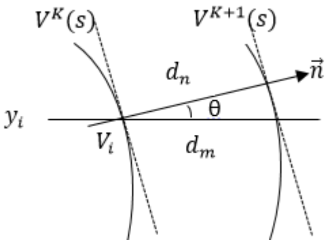


Figure 6. Method for estimating normal front velocity.

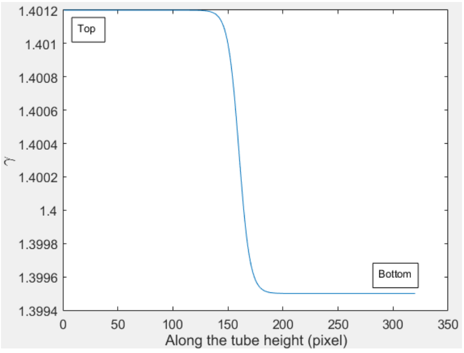


Figure 7. Spatially varying γ for top and bottom of the tube.

By assuming the normal vector $\overrightarrow{n}$ is locally normal to both fronts at K and $K+1$, a normal displacement d_n is estimated as $d_n = d_m * cos\theta$.

The normal displacement d_n is then changed to standard unit of meter with system configuration of $(1px = 0.0002778\ m)$ and the normal velocity V_f is calculated at

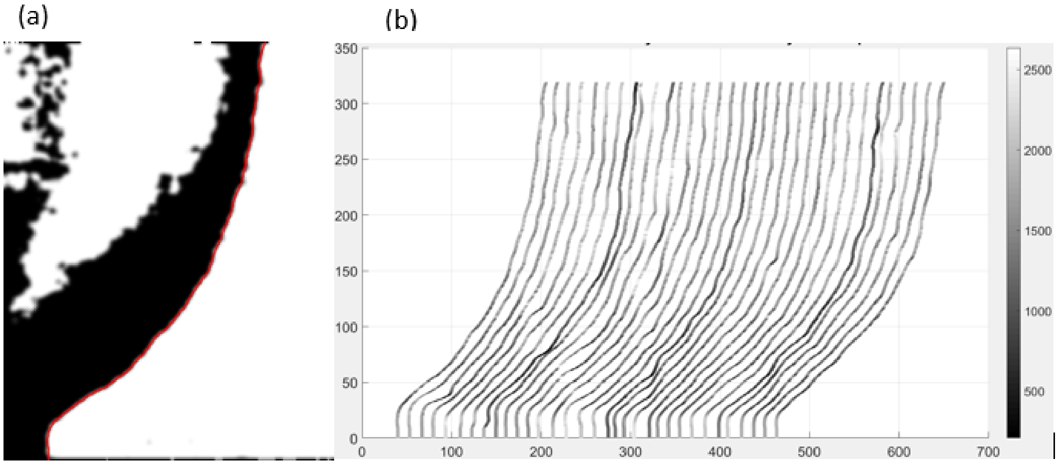


Figure 8. (a)A final contour in filtered binary image.(b) The front contours with local velocity (m/s) given by the color according to the colormap.

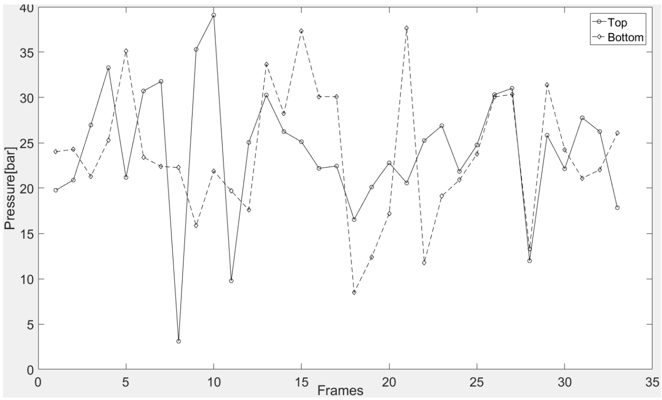


Figure 9. Pressure estimated for top and bottom of each front.

frequency of 500 kHz. By using the calculated velocity to estimate the Mach number M, the pressure behind the front is estimated by normal shock relation shown in (15) (Law, 2010).

$$\frac{P_1}{P_0} = \frac{2\gamma M^2 - (\gamma - 1)}{\gamma + 1}. \tag{15}$$

Here, P_1 is the pressure behind of the wave front and P_0 is the pressure ahead the wave front. The pressure ahead of the wave front is assumed to be 1 bar in this case. The Mach number M is defined as,

$$M = \frac{V_f}{c}, \tag{16}$$

$c = \sqrt{\gamma RT/M}$ is the speed of sound in medium (H_2) at temperature T. R , M and γ are respectively gas constant, Molecular mass and specific heat ratio of H_2. As the gas is different in top and bottom of the tube, γ is taken as spatially varying along the tube from 1.3995 to 1.4012 as shown in Figure 7 (γ for H_2 is 1.4). The speed of sound for hydrogen is calculated and considering temperature T ahead the wave front is uniform at 25^o C. The estimated M is then used for finding pressure P_1 from (15).

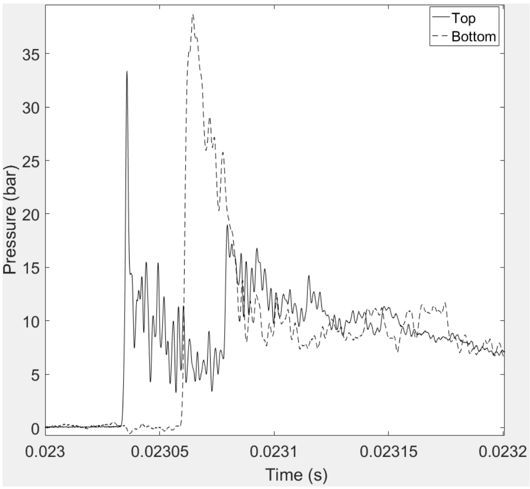


Figure 10. Pressure records from the experiment by the pressure transducers.

6 Results and Conclusion

For clarity, the final contour along the detonation wave front is plotted in filtered image shown in Figure 8(a). The calculated velocities along all the tracked wave fronts from a sequence is plotted in Figure 8(b). As expected the velocity in the upper layer that is filled with combustible gas is relatively higher than the velocity in the bottom layer. Due to the generation of Mach stem along the lower boundary of the tube, velocity of the front around the lower boundary is noticeably high. The average pressure estimated at top and bottom of the each front is plotted in Figure 9 (top and bottom 10 points are used). The reading from pressure transducers mounted on the top and the bottom of the tube at fixed location is shown in Figure 10. Due to the difference between the location of pressure transducers and the moving wave front, direct comparision cannot be justified. Nevertheless, it can be seen that maximum and minimum pressure on both the figure are almost in same range.

DOI: 10.3384/ecp17138174

The designed image processing framework was also used to calculate the velocity for a benchmark experiment. The results of the experiment shows an average detonation velocity of 1967 m/s which is 9 m/s lower than the theoretical value (Gaathaug et al., 2016). Thus, it can be assumed that the framework gained detailed information of the detonation front within expected precision. Further work need to be done considering assumptions made during pressure estimation using a normal shock relation.

References

Tony F. Chan and L.Vese. Active contours without edges. *Image Processing, IEEE Transactions.*, 10:266–277, 2001.

Andre Vagner Gaathaug, Dag Bjerketvedt, and Knut Vaagsaether. Experiments with flame propagation in a channel with a single obstacle and premixed stoichiometric h2-air. *Combustion Science and Technology.*, 182: 1693–1706, 2010.

Andre Vagner Gaathaug, Samee Maharjan, Ola Marius Lysaker, Knut Vaagsaether, and Dag Bjerketvedt. Velocity and pressure along detonation fronts - image processing of experimental results. *Proc. of the Eighth International Seminar on Fire and Explosion Hazards (ISFEH8)*, 2016.

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

Chung K. Law. *Combustion Physics.* Cambridge University Press, New York, USA, 2010.

David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics.*, 42:577–685, 1989.

Stanley Osher and James A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics.*, 79: 12–49, 1988.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics.*, 9:62–66, 1976.

Van-Truong Phama and Thi-Thao Tranc. Active contour model and nonlinear shape priors with application to left ventricle segmentation in cardiac mr images. *Optik - International Journal for Light and Electron Optics*, 127:991–1002, 2015.

Gary S. Settles. *Schlieren and Shadowgraph Techniques: Visualizing Phenomena in Transparent Media.* Springer-Verlag Berlin Heidelberg New York, 2001. ISBN 0.

Chenyang Xu and Jerry L. Prince. Gradient vector flow: A new external force for snakes. *IEEE Proc. Conf. on Computer vision and Pattern Recognigition*, pages 66–71, 1997.

Chenyang Xu and Jerry L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing.*, 71:131–139, 1998.

Zhoua Yan, Ren Shia Wei, Wei Chenc, Yong lin Chenc, Ying Lid, Li-Wen Tand, and Dai-Qiang Chenb. Active contours driven by localizing region and edge-based intensity fitting energy with application to segmentation of the left ventricle in cardiac ct images. *Neurocomputing*, 156:199–210, 2014.

Article II: An Image Processing Framework for Automatic Tracking of Wave Fronts and Estimation of Wave Front Velocity for a Gas Experiment

An Image Processing Framework for Automatic Tracking of Wave Fronts and
Estimation of Wave Front Velocity for a Gas Experiment.
Samee Maharjan, Dag Bjerketvedt and Ola Marius Lysaker.
Published in: Representation, analysis and recognition of shape and motion FroM Image
data, Communications in Computer and Information Science. 187 . Springer, (2018)

An Image Processing Framework for Automatic Tracking of Wave Fronts and Estimation of Wave Front Velocity for a Gas Experiment.

Samee Maharjan*, Dag Bjerketvedt, and Ola Marius Lysaker

Department of process, energy and environmental technology, University College of Southeast Norway, Norway.
email: samee.maharjan@usn.no

Abstract. This work presents an image processing framework designed to automatically track the wave front in a sequence of images from a high speed film. A watershed algorithm is used for segmentation and contouring, while an active contour model is used for controlling the flexibility and the rigidity of the contour. The velocity of the wave front is calculated by estimating the displacement of the front in two frames, divided by the time difference between the frames. The calculated velocity is compared with the sensor measurements. Further, the calculated velocities can be used to estimate thermodynamic properties like the Mach number, the pressure and the temperature across the wave. With the purposed framework, these properties can be estimated along the entire wave front. Otherwise, these thermodynamic properties are limited to either theoretical values or to sparse measurements from sensors. The experiment is done by using a shock tube and the film is captured by a high speed camera using the shadowgraph system.

Keywords: Image processing application, Watershed, Front tracking, Active contour model.

1 Introduction

Image segmentation is one of the important and most explored field in image processing. Among the numerous methods of image segmentation, watershed algorithm which is based on mathematical morphology of an image is well known and widely used, especially for boundary contouring and edge detection [1]. The watershed method takes an image as a topographic surface, where the graylevel values of the image corresponds to the altitude of a surface (minimum being the deepest). The algorithm was first introduced by Digabel and Lantu´ejoul [2] and further modified by Beucher and Lantu´ejoul [3, 4]. Initially, the algorithm was highly suffering by over segmentation due to the large number of local minima present in an image. To overcome with the problem of over segmentation, Meyer and Beucher proposed a strategy known as marker-controlled segmentation [5].

More on the application of watershed method in grayscale images and overall review of the marker controlled segmentation can be found in [6, 7].

The active contour or snake model was first developed by Kass et al. in 1988 [8]. The snake model is an essential part in image processing and computer vision applications, mainly used for shape modeling [9] and motion tracking [10]. A snake is a moving curve within an image, which eventually lie itself around the surface/edge of the desired object. For the present work, an open contour model developed in [11] is used.

In general, shock waves appear in many flow fields, for example, at the wings of an airplane, in explotions, fired bullets, exhaust of engines etc. The study of the structure and the properties of a shock wave and its boundary layer interactions has been the point of interest during World War II [12, 13]. Shock wave initiates when the speed of the wave exceeds the speed of sound in the meduim. It is characterized by the fact that across the shock regions, the gas properties like the pressure, the temperature, and the gas density increases drastically. The major cause for shock initiation is detonation, but sources could also be explosion or lighting. The study of shock waves is a key component in the field of aerospace and oil and gas industry as it could provide necessary information for designing better aircraft and safety equipment. Generation of a shock wave through detonation and its properties can be studied in [14].

Image processing technology has become a valuable tool in the field of gas combustion and fluid mechanics. However, most of the time, the use of images are limited to visualize the structure of the waves [12]. There is a lack of an image processing framework which manage to extract information of the wave properties. One challenging task is to capture this extremely high-speed phenomena. The generated waves propagate at a speed of 200 m/s and above, and the reactions of interest are completed within a microsecond. Therefore, a special high-speed camera designed for capturing these phenomena is needed. In addition, the high speed images from a gas experiment can vary rapidly from frame to frame due to the continous chemical reaction, which demands a robust framework. In the past, a few work has been done based on numerical simulation and thermodynamic differential equations [15]. However, to obtain a tolerable accuracy using numerical models is a CPU demanding task. The main goal of the paper is to design a complete image processing framework, which pre-process/filter the images and automatically track the wave front in these images. The tracked fronts are then used to estimate the velocity of the wave.

The rest of the paper is organized as follows. Section 2 gives a brief description on the experimental setup for the generation of a shock wave. The methodology behind image filtering, front tracking and post processing is described in Section 3, which is followed by Section 4 in which the procedure for calculating velocity from the tracked fronts is presented. The results from the framework are shown in Section 5. Lastly, the conclusion and some possible further work is discussed in Section 6.

2 Experimental set up

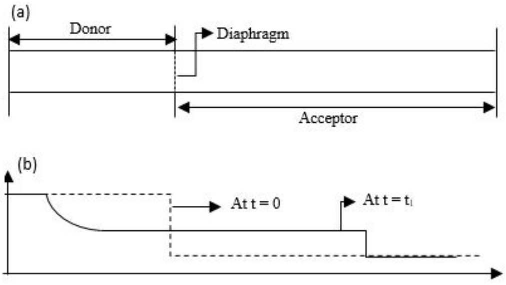


Fig. 1. (a) Schematic representation of the shock tube (b) pressure distribution along the tube at t = 0 and t = $t_1 \neq 0$.

The shock wave was generated in the laborotory using a pure CO_2 gas and a shock tube, with the initial pressure and temperature of 10kPa and 274 K respectively. A shock tube is one of the most used and essential laboratory instrument for study of the fluid mechanics and gas combustion [16]. A shock tube typically is a closed tube that consists of two chambers, one with a high pressure known as the donor section and another with fairly low pressure known as the acceptor section. Fig. 1 shows the schematic representation of a shock tube with its pressure distribution. The shock propagation is captured at the closed end of the acceptor section. Due to the boundary layer following behind the incoming shock, the reflected shock wave bends around the boundary creating an oblique shock. An oblique shock makes a certain angle with the boundary of the tube known as a shock angle. Hence, a single reflected shock wave can be divided into two parts: a normal shock and an oblique shock (see Fig. 2). This paper focus on the reflected shock wave, hence the notation 'shock wave' refers to the reflected shock wave onwards.

Some of the images from the high speed film capturing the shock wave propagation are presented in Fig. 2. The growing of the oblique shock is clearly visible in the images. A shadowgraph system [17] is used for capturing the wave propagation.

3 Methodology

In this section, the four parts of the image processing framework are described. For simplicity, only the lower halves of the images are taken for further processing as the phenomena is almost identical at both the top and the bottom boundary.

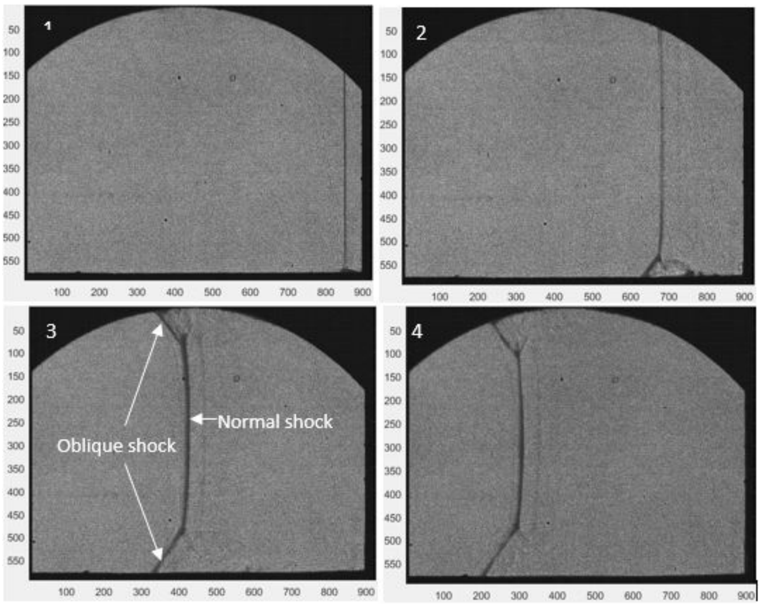


Fig. 2. Some of the raw images from the high speed film showing the reflected shock wave propagation. The shock wave travels from left to right, as shown in frames 1-4.

3.1 Pre-processing

The pre-processing of an image not only reduce the variations of noise present in the image, but also reduce the problem of over segmentation of an image. The pre-processing framework presented here consists of 3 steps. The first step is the background subtraction. An image was formed by making an average image based on all the images prior to the front propagation, and this image was then successively subtracted from all the images with a visual front. The background subtracted images are normalized to intensity level $[0-1]$ in the second step. The third step is filtering of noise from the normalized image. The filtering is done by using the log transformation of the image followed by the $[2 \times 2]$ median filter [18].

3.2 Segmentation

The filtered image is changed into a binary image by using Otsu method [19] followed by the morphological operation 'closing'. The closing of the binary image [5] enhanced the edges of an object as well as filling the tiny gaps found close to the edges. The sequence of images in Fig. 3 shows the output from each step of the pre-processing operation of an image from the high speed film.

Generally, the watershed segmentation is not performed directly to the original image due to over segmentation. In this work, the distance transformed

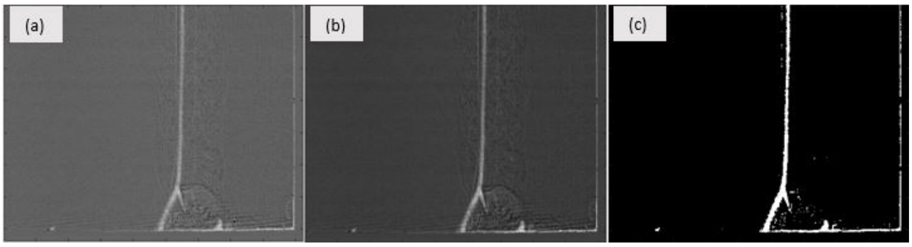


Fig. 3. (a) Background subtracted normalized image (b) filtered image (c) thresholded binary image.

image [20] is used for the segmentation. The distance transformed image is suitable when the edge of the object is diffuse, but visuable. Fig. 4(a) and Fig. 4(b) respectively shows the distance transformed image and the output of the watershed transformation of Fig. 3. The watershed algorithm uses 8 neighbourhood pixels for the catchment building and consecutive flooding [21].

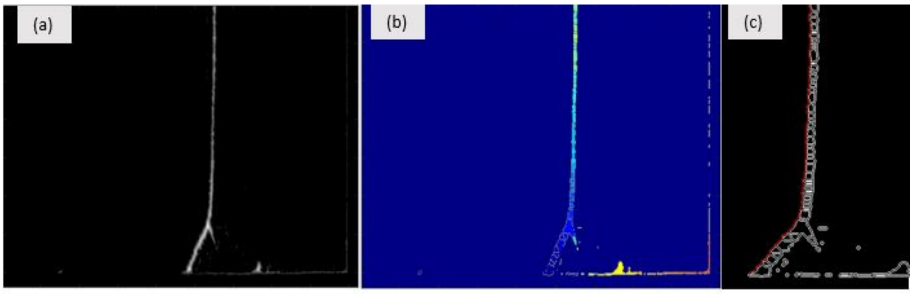


Fig. 4. (a) Distance transformed image (b) watershed output (c) morphologically removed image with initial contour.

3.3 Front tracking/Contouring

To represent of the front in the segmented image, the morphological operation 'remove' is done. This operation removes all the internal pixels in an object, but keep all the edge pixels unchanged [22]. By using a priori information of the direction of the wave propagation, a contour point is placed at the first position where the intensity value changes from 0 to 1. The search starts from the left to the right i.e. opposite direction of propagation. The points are tracked for all the rows and the contour is created by simply joining them. If the size of an image is $[m, n]$, the contour can now be represented as a vector of size $[m, 1]$. The initial contour plotted in the morphologically removed image is shown in Fig. 4(c).

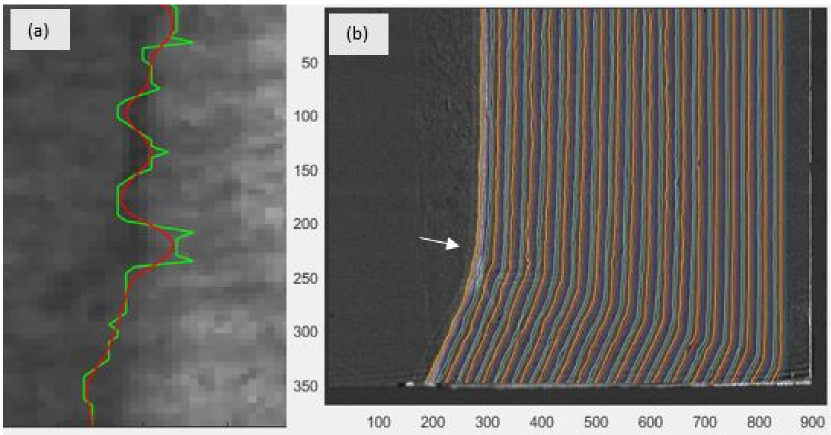


Fig. 5. (a) This subsection of the front is selected from the area pointed with a white arrow in (b), initial contour (green) and final smoothed contour (red). (b) final fronts from the high speed film tracked by the framework.

3.4 Post processing

It can be seen from Fig. 5(a) that the initial contour (green) has a ragged shape. Thus, a proper module for smoothing the curve was needed. By using a snake model [11], the smoothing was done locally using only on a few neighbouring points. A basic active contour model or snake model defined in [8] is a controlled continuity parametric curve, formed within an image, where the snake is initialized manually by set of x-y points. However, for this work the contour from the watershed segmentation is taken as an initial snake, eliminating any human interactions. By representing the position of the snake parametrically as $P(s) = (x(s), y(s))$, its energy function is defined as in (1)

$$E_{snake} = \int_s E_{int}(P(s)) + E_{ext}(P(s))ds. \tag{1}$$

The first term in (1), E_{int} is an internal energy and the second term, E_{ext} is the external energy function. An open active contour model developed in [11] which is used to smooth the curve. Fig. 5(a) shows the contour from watershed (green) and the smoothed contour (red) by the snake model. It also shows convergence of the contour to the actual wave front around the pointed part in Fig. 5(b)). Fig. 5(b) shows the 100 final fronts of the shock wave tracked by the designed framework in the sequence of images of a high-speed film.

4 Velocity calculation

By estimating the displacement of the front in two frames combined with the framing frequency of a camera (which will give the time difference between the

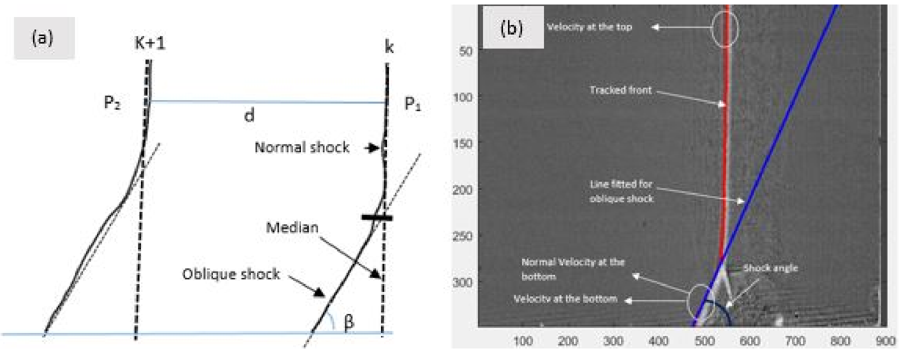


Fig. 6. (a) Schematic sketch of two consecutive fronts with the fitted lines, the median and the shock angle. (b) A raw image with actual tracked front and fitted line for the oblique shock.

frames), the velocity can then be calculated. For the present work, the frequency of a camera is 500 kfps and the pixel to meter scaling is approx. $1px = (0.0001 \pm 5\%)$ m. The velocity of a front at point $P_1 = (x_1, y_1)$ (x_1 and y_1 represents row and column) in image k can be estimated by using the point $P_2 = (x_1, y_2)$ of the front in the consecutive image $k + 1$. If a displacement is $d =\mid P_2 - P_1 \mid$, then velocity v will be,

$$v = \frac{d \times 0.0001}{0.2 \times 10^{-5}} m/s. \tag{2}$$

One of the raw image from the film with the tracked front (red) and the line formed for the oblique shock (blue) is shown in Fig. 6(b). The shock angle β can be estimated by forming a line (equation) for the oblique shock, therefore the contour needs to be divided into two sections; normal shock (upper) and oblique shock (lower). For this, a median of top 100 points of the tracked front (red) is used. The separating point is chosen to be situated where the front starts to deviate from a median in an increasing order. The equation for the oblique shock is formed by using all the points below the separating point shown in Fig. 6(b) by a blue line. Fig. 6(b) also point out the desired calculations from the tracked front and the fitted line. The top and bottom velocities are calculated by taking 10 points on each of them (highlighted by an oval structure). The normal velocity is then estimated by using velocity at the bottom v and shock angle β i.e. $vsin\beta$. The calculated velocity is then used to estimate the Mach number at that point. A Mach number M is the ratio between the velocity of the wave and the speed of sound in the medium. By using the Mach number along with the initial temperature and pressure, the final pressure and temperature can be estimated with the use of shock relationships. More about the Mach number, its importance and implementation along with the shock relationships can be found in [23].

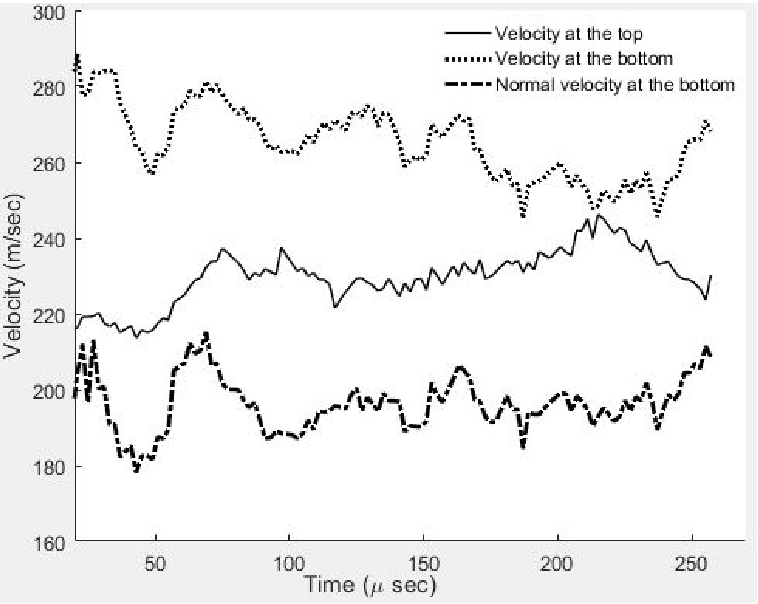


Fig. 7. Average velocity calculated for each frame at the top , bottom and average normal velocity at the bottom plotted along the time.

5 Results

The velocity at the top and the bottom of the front along with the normal velocity at the bottom as pointed in Fig. 6(b) are plotted in Fig. 7 for each front. Please notice that the time start from the first frame considered and is estimated from the frame rate of the camera which gives 2 μsec per frame.

The calculated average velocity is compared with the experimental results that was calculated from the readings of the pressure transducers. The method of calculating velocity and other thermodynamic properties from the pressure transducers and all the experimental results will be published in a separate publication (in progress). The results are also compared to the similar work done by using pattern matching and segmented regression in [24]. The overall comparison is shown in Table 1.

The comparison shows that the designed framework produces results within a necessary precision. The difference between the experimental result (pressure transducer readings) and the result from the framework is 11 m/s (top) and 1 m/s (bottom). The error of the framework due to misplacement of the contour by 1 pixel leads to deviation of approx. 5.23 m/s. Even though, individual velocity along the front may vary with the experimental value, overall average seems to match well.

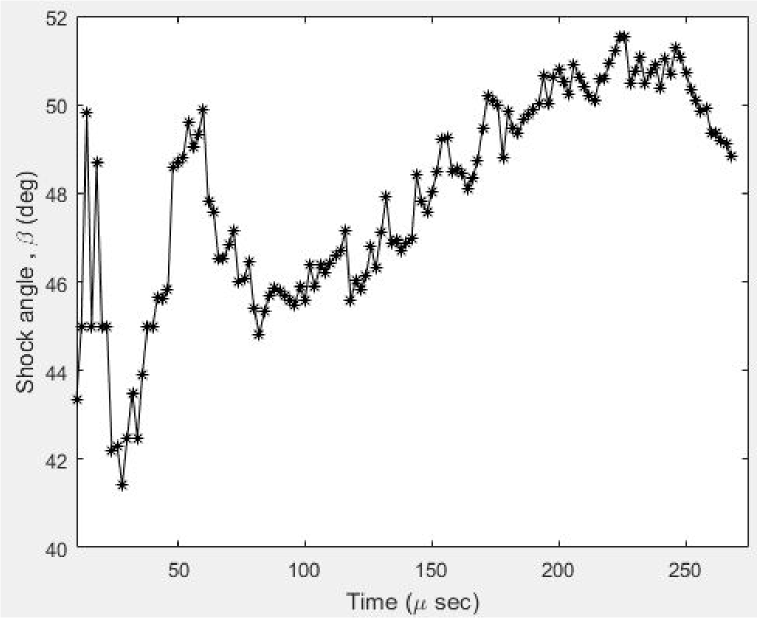


Fig. 8. Shock angle (β) at the foot of all the tracked fronts plotted along the time.

6 Conclusion and Further work

From the results presented in Section 5, it can be concluded that the designed framework produces results in accordance with the pressure transducer measurements. Furthermore, the velocity at each point along the front can now be calculated by using the procress described in Section 4. The procedure and results of estimating thermodynamic properties of gas like the Mach number, the pressure and temperature ratios by using this framework will be presented in a different publication [25].

Table 1. Comparison between results from different methods.

Exp.no		Average velocity at the top	Average velocity at the bottom	Av. Shock angle (degree)
2516	Pressure transducer results	216 m/s	266 m/s	48
	Results from [24]	229 m/s	262 m/s	50
	Results from the designed framework	227 m/s	267 m/s	48.9

References

1. Beucher, S., Bilodeau, M.: Road segmentation and obstacle detection by a fast watershed transformation. In: Intelligent Vehicles '94 Symposium, Proceedings of the, pp. 296–301.(1994)
2. Digabel, H., Lantuejoul, C.: Iterative Algorithms. In: Proceedings of the 2nd European Symposium Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, pp. 85–89.(1978)
3. Beucher, S., Bilodeau, M.: Use of watersheds in contour detection. In: International workshop on Image Processing: Real-time edge and motion detection/estimation, pp. 17–21. France (1979)
4. Beucher, S., Centre De Morphologie Mathmatique.: The Watershed Transformation Applied To Image Segmentation. In: Scanning Microscopy International, pp. 299–314.(1991)
5. Beucher, S., Meyer, F.: Morphological segmentation. J. Vis. commun. Image Represent. 1, 21–46 (1990)
6. Roerdink, J.B.T.M., Meijster, A.: The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. In: Fundamenta Informaticae, pp. 187–228. IOS Press,(2001)
7. Vincent, L., Soille, P.: Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. In: IEEE transactions on pattern analysis and machine intelligence, pp. 583–598.(1991)
8. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Comput. Vision. 1, 321–331 (1988)
9. Terzopoulos, D., Fleischer, K.: Deformable models. The Visual Computer. 4, 306–331 (1988)
10. Peterfreund, N.: Robust Tracking of Position and Velocity With Kalman Snakes. In: IEEE transactions on pattern analysis and machine intelligence, pp. 583–598.(1999)
11. Maharjan, S., Gaathaug, A.V., Lysaker, O.M.: Open Active Contour Model For Front Tracking Of Detonation Waves. In: Proceedings of the 58th Conference on Simulation and Modelling, pp. 174–179. Linköping University Electronic Press, Sweden (2017)
12. Mark, H.: The Interaction of a Reflected Shock Wave With the Boundary layer in a shock tube. Cornell University, Ithaca, New York, (1958)
13. Courant, R., Friedrichs, K.O.: TSupersonic Flow and Shock Waves. Interscience Publishers, New York, (1948)
14. Nikiforakis, N., Clarke, J.F.: Numerical Studies of the evolution of Detonations. Mathl. Comput. Modelling. 24, 149–164 (1996)
15. Glimm, J., Grove, J.W., Li, X.L., Zhao, N.: Simple Front Tracking. In: Contemporary Mathematics, pp. 133–149. (1999)
16. Glass, I.I., Patterson, G.N.: A theoretical and experimental study of the shock tube. Journal of the Aeronautical Sciences. 22, 73–100 (1955)
17. Settle, G.H.: Schlieren and Shadowgraph Techniques: Visualizing Phenomena in Transparent Media. Springer-Verlag Berlin Heidelberg, New York, (2001)
18. Lim, J.S.: Two-Dimensional Signal and Image Processing. Prentice-Hall, Inc., USA, (1990)
19. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics. 9, 62–66 (1979)
20. Paglieroni, D.W.: Distance transforms: properties and machine vision applications. CVGIP: Graphical Models and Image Processing. 54, 56–74 (1992)

21. Meyer, F.: Topographic distance and watershed lines. Signal Processing. 38, 113–125 (1994)
22. Gonzalez, C.R., Woods, E.R.: Digital Image Processing. Prentice-Hall, New Jersey, USA, (2000)
23. Law, C.K.: Combustion Physics. Cambridge University Press, New York, USA, (2010)
24. Siljan, E., Maharjan, S., Lysaker, M.: Wave Front Tracking using Template Matching and Segmented Regression. In: Proceedings of the 58th Conference on Simulation and Modelling, pp. 326–331.Linköping University Electronic Press, Sweden (2017)
25. Maharjan, S., Bjerketvedt, D., Lysaker, O.M.: Estimation of Mach Number and Wave Properties of a Reflected Shock Using Image Processing. *In process*,.

Acknowledgment

All the experiments were conducted at California Institute of Technology in collaboration with University College of Southeast Norway.

Article III: Wave Front Tracking in High Speed Videos Using a Dynamic Template Matching.

Article III: Wave Front Tracking in High Speed Videos Using a Dynamic Template Matching.

Samee Maharjan.

Published in: 9th Iberian Conference on Pattern Recognition and Image Analysis, Lecture Notes in Computer Science book series (LNCS, volume 11867). Springer, (2019)

Wave Front Tracking in High Speed Videos Using a Dynamic Template Matching.

Samee Maharjan

Department of process, energy and environmental technology, University of
South-Eastern Norway (USN), Norway.
email: samee.maharjan@usn.no

Abstract. In recent years, image processing has been evolving as an important tool to estimate the properties of a generated wave during a gas experiments. The wave properties were estimated by tracking a wave fronts in the images of a high speed video, captured during the experiment. In this work, we purposed a dynamic template matching method, based on the mean square error (MSE) between the intensity values of a template and its foot print in an image to track the wave front. At first, a dynamic template of a predefined size [5 × 20] was created, whose values varies according to the minimum and maximum intensity of the considered image. Secondly to reduce the processing time, a bounding box was set around the area of interest in the considered image, such that the matching process is limited within the area covered by the bounding box. The purposed method was tested for four different high speed videos from four different gas experiments conducted two different experimental set-ups. All the results from the purposed method are within an acceptable accuracy.

Keywords: Image processing application, Dynamic template, Front tracking.

1 Introduction

Template matching or pattern matching is an image processing method to detect the desired object in an image by using a predefined template. A template matching starts with creating a template of relatively smaller size whose one or multiple features matches with the features of the desired object. Then, the created template is slid in a pixel-by-pixel basis, computing the similarity between the template features and its footprint in the image [1]. The few common features that are used for calculating a similarity while matching are, normalized cross correlation (NCC), the sum of absolute difference (SAD), the sum of squared error (SSD), mean square error (MSE) [2]. In this work, we took the template matching based on MSE, due to its simplicity and fast processing. The MSE takes a mean of the squared difference between the intensity of each pixel in the template and the corresponding pixel in its footprint in the image. Assuming template T of size [m × n] slides over an image I, then at each position (x,y) in I, MSE is estimated as in (1). The template matches the best in the image pixel where MSE is the minimum.

$$MSE(x,y) = \frac{1}{no.of pixel} \sum_{k=1}^{m} \sum_{e=1}^{n} [I(x+k, y+e) - T(k,e)]^2. \tag{1}$$

In general, a gas explosion can be defined as a process where combustion of a premixed gas cloud, i.e. fuel-oxidiser is causing a rapid increase in pressure in a close vessel or confined area due to any kind of external energy. When the pressure exerted inside the vessel is higher than the vessel can hold then it explodes producing extremely powerful and destructive waves [3]. The most common approach to study these geberated waves and estimate their characteristics are by using the pressure transducers in the experimental area and/or computer simulations [4]. However, in recent years, the estimation of wave characteristics using the high speed videos and image processing have been emerging [5] - [7]. A high speed video captured during the gas experiment was later processed using image processing technologies to track the wave front in the images of the video. By tracking the wave front in these images, the wave characteristics like speed, temperature and pressure were estimate.

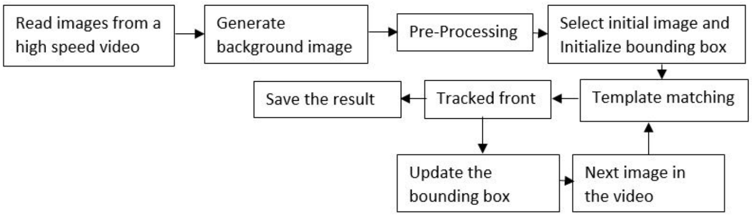

Read images from a high speed video
Generate background image
Pre-Processing
Select initial image and Initialize bounding box
Template matching
Tracked front
Save the result
Update the bounding box
Next image in the video

Fig. 1. Block diagram of the framework.

However, due the poor quality of images in a high speed videos, most of the image processing framework consists of multiple processing units, which not only cost a computational time but also require manual interference. One of the previous work that used template matching for front tracking can be read [8]. The tracking was done in the raw images with two predefined templates: one for a straight wave and one for the tilted (oblique wave) in a single image (refer to Fig. 2 second row). The framework then use a post-processing for finding the optimum result from both templates. The matching was based on the predefined average intensity of the template for an individual video. Even though, the framework works fine with some manual input, the sliding of a template in an overall image along with post-processing takes a high computational time. In this work, we purpose a new method of creating a dynamic template and sliding the created template within a certain region in an image bounded by a bounding box. As, we are working with a set of images from a multiple high speed videos, whose intensities varies with each image, we create a template with a fixed size but with a varying value depending in the intensity of each image to be matched.

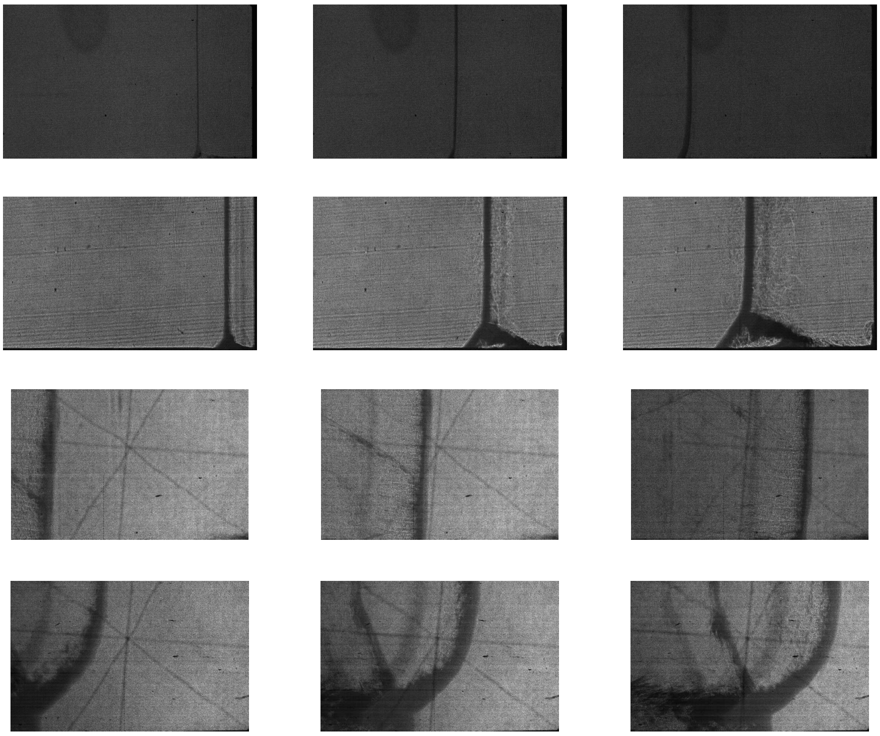


Fig. 2. Some of the images from different gas experiments chronologically sorted from left to right. An arrowhead points towards the direction of the wave propagation in the corresponding video. From top to bottom row: N_2, CO_2, H_2 and $H_2 + air$ experiment.

The rest of the paper is organized as follows. Section 2 gives a brief description on the high speed videos and the methodology of template creation, setting bounding box and matching the created template in the area surrounded by the bounding box. The results from the purposed method with some discussion are presented in Section 3. Lastly, the conclusion and some possible further work is discussed in Section 4.

2 Materials and Methods

The block diagram presented in Fig. 1 depicted the work flow of the overall framework. More on the each part are described in following sub-sections.

2.1 High speed videos

The four high speed videos captured during four different gas experiments were processed in this work. The experiments were conducted with N_2 (Exp.2534), CO_2 (Exp.2558), H_2 (Exp.0016) and $H_2 + air$ (Exp.0022) gases. Figure. 2 shows the wave propagation during the considered four gas experiments. The images in each row are from an individual experiment and are presented in chronologically order from left to right. All the upcoming figures in this paper with four rows follows the structure row wise i.e. from top to bottom row: N_2, CO_2, H_2 and $H_2 + air$ experiment. An arrowhead points towards the direction of the wave propagation in the corresponding video. The characteristics of the generated wave during any gas explosion depends on the characteristics of the gas itself and the initial conditions like initial pressure and temperature. Due to this reason, the structure of wave is different in each experiments. The objective of this work is to design a single framework, which track the wave front in all these images/videos regardless of their structure.

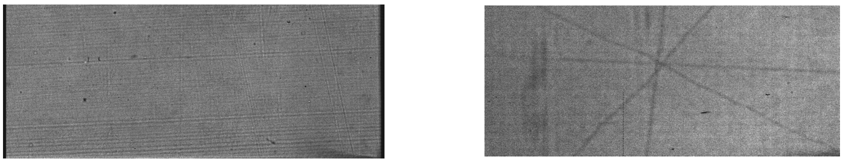


Fig. 3. Background image created for left: CO_2 and right: H_2 experiments.

The first two experiments were conducted in a shock tube and the propagating wave is known as a reflected shock wave. The details of a shock tube and the method of capturing high speed video while experimenting in a shock tube can be read in [5]. The bottom two experiments were conducted in an open end experimental tube and the propagating wave is known as a detonation wave [10]. A schlieren technique [12] of imaging was used for capturing all the videos, and a special high speed camera called 'Kirana' was operated with the frequency of 500,000 frames per second. Each high speed video consists of 180 images of size [768 × 924] pixels $\approx$ [70 × 100] mm. However, some part of the image contains capturing window, which was cropped appropriately before processing. The final size of images for respective experiments are : N_2 and CO_2 is [400 × 910], H_2 is [356 × 631] and $H_2 + air$ is [356 × 640].

2.2 Pre-processing/Filtering

Due to the high frame rate of the camera and the ongoing chemical reactions during the experiment, the images in the high speed videos are generally with high amount of noise. Therefore, it is preferred to pre-process the images before further processing. The pre-processing of an image in the current framework consists of a background subtraction followed by a low pass frequency filtering. For background subtraction, a background image was created by taking average intensity values of the initial images which doesn't consists of the visual wave front. The background image created for the high speed videos of CO_2 and H_2 experimnets are shown in Fig. 3. After that, the created background image was subtracted from each image with the visual front or which needs to be procressed. The remaining noise in the background subtracted image was then removed by using a low pass filter with the threshold frequency of 50 Hz. The examples of the background subtraction in the images from high speed videos is presented in Fig. 4 left column, and the right column shows their respective low pass filtered images. The process of the frequency filtering in the images and its benefits can be read on [15].

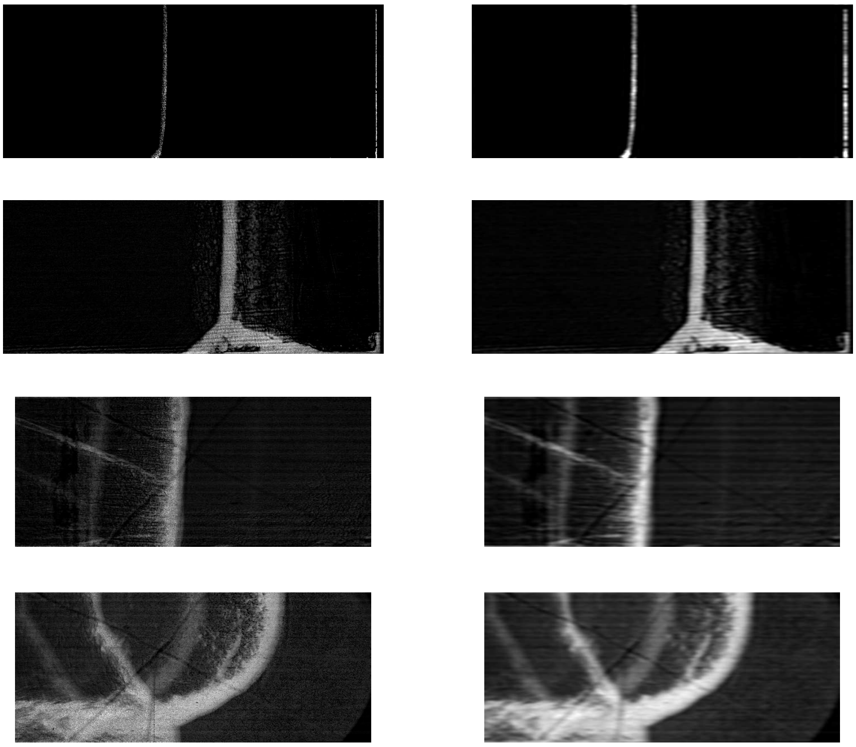


Fig. 4. Result of pre-processing, right column: result after background subtraction, left column: respective filtered image after LPF.

2.3 Template creation

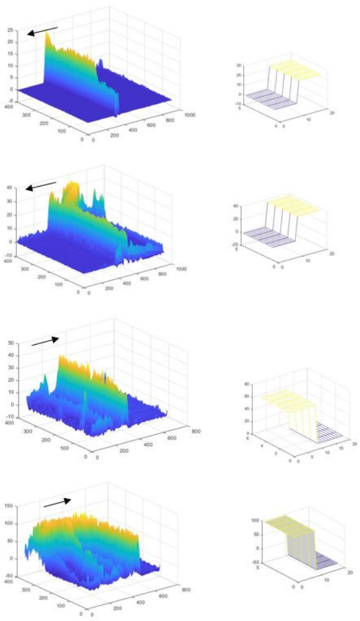


Fig. 5. The mesh plot showing the intensities. Left : the filtered images and right: a template created for the corresponding images (right).

The close look of the intensity difference between wave band and background in the background subtracted images suggested that the waveband contains of bright pixels where as background contains of dark pixels. With this reference, a dynamic template of predefined size [5 × 20] was created, which values depends on the minimum and maximum intensity of the considered image. One half of the template, contains the minimum intensity value which should technically be the intensity of a background, while another half contains the maximum intensity value which should technically be the intensity of the wave band. However, which side of template takes the minimum and maximum value depends on the direction of the wave propagation. For example, for first two videos in which the wave is propagating from right to left, the wave front is located at the left side of the wave band. Therefore, the right half of the template contains the maximum intensity value (wave) and left half contains minimum (background). The values in the template for the bottom two videos would be exactly opposite as the wave is propagating in an opposite direction and the wave front lies at the right side of the wave band. The mesh plot illustrating the intensity level of a wave band and background in the filtered images from Fig. 4 along with the mesh plot of their respective created templates are presented in Fig. 5.

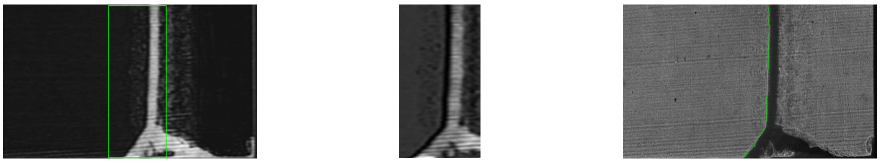


Fig. 6. Template matching process left: a filtered image from CO_2 experiment with a bounding box for matching, center : MSE calculated from matching within the bounding box, right: the result of choosing the minimum MSE in each row presented by the green curve in a raw image of the same image in left.

2.4 Bounding Box

As it can be seen in all the images presented in previous sections, the wave actually stands in the small part of image. Hence, sliding a template all over the image will only increase the processing time. Therefore, to minimize the processing time, a bounding box was created around the area of interest (wave) such that the sliding of template occurs only inside the bounding box. The height of bounding box is always the number of rows in the image, whereas the width varies depending on the horizontal span of wave band in the image. For the initial image, the bounding box was set at the side of image where, the wave originates and then it moves along the direction of wave propagation with each consecutive images. The movement of the bounding box was governed by the values of the previous tracked front.

Lets take an example of the CO_2 experiment, the bounding box was set initially at the right end of the image where the wave originated. The right end of the bounding box was set at the right end of image itself, whereas the left side was set 200 columns ahead of the right end. After the first front was tracked, the position of the bounding box was updated according to the position of the tracked front. As the wave was always going forward (towards left), the right side of the bounding box was updated with the median value of the first tracked front while keeping the width of the bounding box constant. The choice of 200 columns as the width of the bounding box came from the a priori visualization of the high speed video which gave rough idea about the total wave span (refer to Fig. 2 second row). Similarly for the H_2 experiment, the bounding box with the width of 100 was initially set at the left side of the image. After first tracking, the left side of the bounding box was then updated with the median of the first tracked front keeping the width as it is. However, in $H_2 + air$ experiment to accommodate the shape of the wave, the bounding box of width 200 was set with the median value of last 50 rows. One of the raw images from CO_2 and H_2 experiment are presented in the left image of Fig. 6 and Fig. 7 respectively. The width of bounding box for the N_2 experiment are 50.

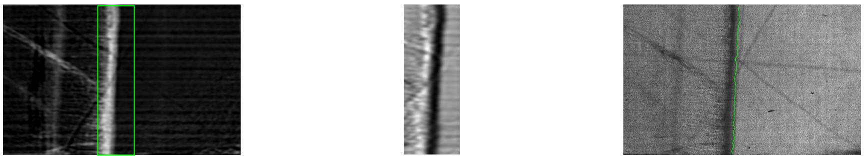


Fig. 7. Template matching process left: a filtered image from H_2 experiment with a bounding box for matching, center : MSE calculated from matching within the bounding box, right: the result of choosing the minimum MSE in each row presented by the green curve in a raw image of the same image in left.

2.5 Template Matching

The sliding of the template always started from the top left of the bounding box and moved towards top right. For example, a bounding box in Fig. 6 spreads from top to bottom (all rows) and column 350 to 550. The first matching took place at top left of bounding box i.e. 1^{st} row and 350^{th} column of the image and the template slide along each column calculating MSE at each pixel till 531^{th} column. After, it reached 531^{th} column, it slide to 2^{nd} row 350^{th} column and continued till 531^{th} column and so on. Four pixels at the bottom and 19 pixels at left of bounding box was exempted due to boundary adjustment. At each position, MSE between the template and its footprint in the image was estimated as in (1), such that $x = 1 : 396, y = 350 : 531$. After completing calculating MSE in one image, a pixel with the least MSE was picked out in each row. Please note that the actual front position is in the middle of the template, such that the minimum MSE pixel in any row gives the front position at 2 rows below. The position of the wave front is then 10 columns behind actual column of MSE pixel. For example, the minimum MSE point in a 1^{st} row is at lets say column y_m actually gives the front position in 3^{rd} row which will be in column $y_m + 10$. Hence, there was no front tracked for top 2 and bottom 2 rows.

3 Results and Discussion

A created bounding box of width 200 pixels, the result of template matching within the bounding box and the position of front in one of the raw image from CO_2 experiment is shown in Fig. 6. Similarly, the image from H_2 experiment with a bounding box of width 100 pixels, the calculated for the bounding box and result from the minimum is presented in Fig. 7. Figure. 8 summarizes the results of a dynamic template matching based on MSE for tracking wave fronts in four different high speed videos. The first and second column presents the result in an individual images while, third columns shows all the tracked fronts with their respective position in the image. High speed video of N_2 and CO_2 are

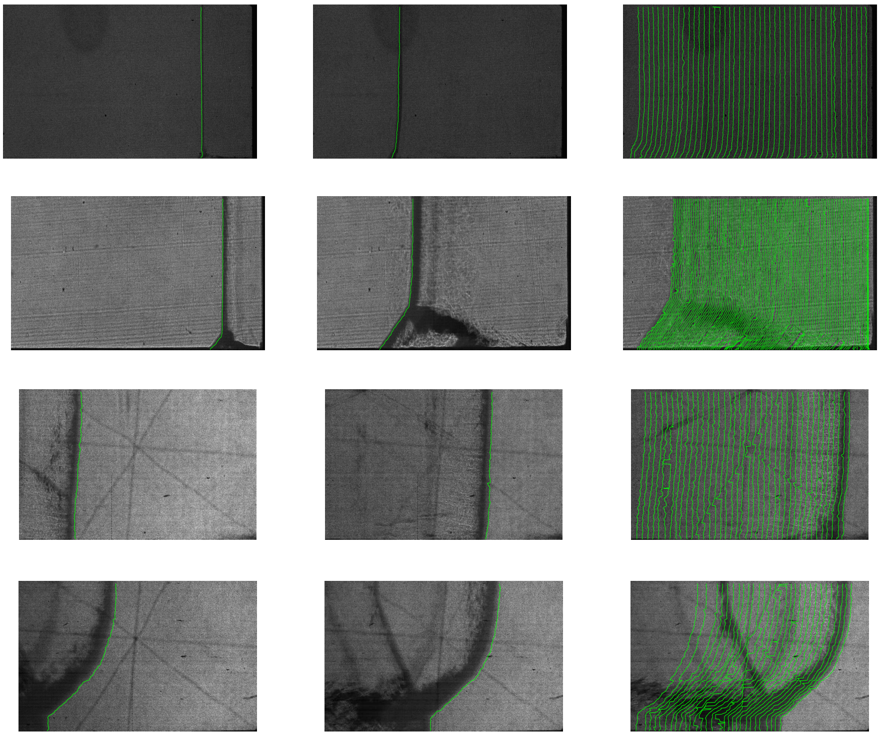


Fig. 8. Template matching process left: a filtered image from H_2 experiment with a bounding box for matching, center : MSE calculated from matching within the bounding box, right: the result of choosing the minimum MSE in each row presented by the green curve in a raw image of the same image in left.

comparatively with less background noise than the $H_2 + air$ and H_2 experiment. The tracked fronts are therefore with less or no distortion as seen in top two rows. However, few distortions can be seen in the results from $H_2 + air$ experiment, the noise within the bounding box is matched more than the actual front. In such cases, some post processing should be performed, as simple one can be the smoothing of the front or piecewise line fitting.

The framework/matching process also tested in the raw images as well as the background subtracted images from the high speed videos. For better illustration in Fig. 9, one image from each experiment are presented in a raw, background subtracted and filtered form with the respective front tracked by the method in the same images. For better quality video with N_2, the results are almost same for all type of images. For the H_2 experiment, the results are smoother and better with each step of pre-processing. In contrast, for $H_2 + air$ and CO_2 the worst results is while using background subtracted image as the noise in the background subtraction enhanced a background noise as line structure inside the wave band.

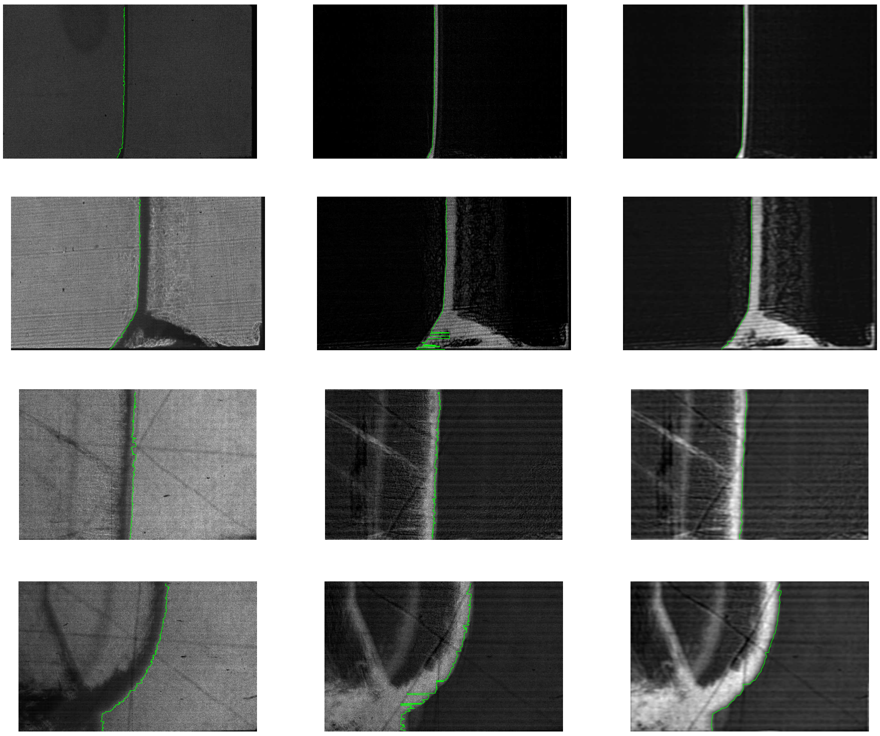


Fig. 9. Template matching process left: a filtered image from H_2 experiment with a bounding box for matching, center : MSE calculated from matching within the bounding box, right: the result of choosing the minimum MSE in each row presented by the green curve in a raw image of the same image in left.

Figure 10 shows the result of using 'prewitt' edge detection method from Matlab image processing toolbox corresponding to right bottom image in Fig. 4. Some of the available edge detection methods in various processing toolboxes were able to detect the edges, however they did not provide the required precision of front position. The red curve is plotted with the first white pixel from right and green one is from the template matching. This shows the importance and the advantage of using a robust method like template matching in order to track the exact front position.

4 Conclusion

A dynamic template with the predefined size and varying value was created to track the wave front the high speed videos. A bounding box which size varies with the size of the wave in the image was set in the images for sliding the created dynamic template to minimize the processing time. The use of bounding box has minimize the processing time by more than 1/3 times. The time of execution of one image from H_2 is 0.440 seconds while using bounding box while took 1.839 seconds while not using bounding box. However, the setting and the movement of of the bounding box depends upon the wave propagating direction and structure of the wave, so a prior information about the structure of the wave in an image is necessary. Visually, all the results are within a acceptable accuracy and a purposed method of template matching have a huge potential for tracking various kinds of wave. The purposed method can also be used without any pre-precessing, however, the tracked fronts would be rougher hence, some post processing are suggested.

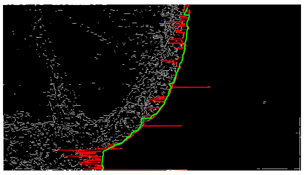


Fig. 10. The result of using 'prewitt' edge detection method corresponding to right bottom image in Fig. 4. Red curve - first white pixel from right and green curve - the template matching.

References

1. Brunelli, R. : Template Matching Techniques in Computer Vision: Theory and Practice. Wiley Publishing,(2009)
2. Ouyang, W., Tombari, F., Mattoccia, S., Stefano, L.D, Cham, W.K : Performance Evaluation of Full Search Equivalent Pattern Matching Algorithms. In: IEEE Transactions on pattern Analysis and Machine Intelligence, 34 (1), 127–143 (2012)
3. Bjerketvedt, D., Bakke, J.R., Wingerden, K.V. : Gas explosion handbook. In: Journal of Hazardous Materials, 52 , 1–150 (1997)
4. Tseng., T. I., Yang., R. J.: Simulation of the Mach reflection in supersonic flows by the CE/SE method. In: Shock Waves,14 (4), pp. 307–311. (2005)
5. Damazo, J.S. : Planar Reflection of Gaseous Detonations. Doctoral Thesis, California Institute of Technology, Pasadena, California (2013)

6. Maharjan, S., Gaathaug, A.V., Lysaker, O.M.: Open Active Contour Model For Front Tracking Of Detonation Waves. In: Proceedings of the 58th Conference on Simulation and Modelling, pp. 174–179. Linkoping University Electronic Press, Sweden (2017)
7. Maharjan, S., Bjerketvedt, D., Lysaker, O.M.: An Image Processing Framework for Automatic Tracking of Wave Fronts and Estimation of Wave Front Velocity for a Gas Experiment. In: Representation, analysis and recognition of shape and motion FroM Image data, CCIS. 187 . Springer, (2018)
8. Siljan, E., Maharjan, S., Lysaker, O.M.: Wave Front Tracking using Template Matching and Segmented Regression. In: Proceedings of the 58th Conference on Simulation and Modelling, pp. 326–331. Linkoping University Electronic Press, Sweden (2017)
9. Akbar, R.: Mach reflection of gaseous detonations. California Institute of Technology, Pasadena, California (2013)
10. Gaathaug, A.V., Maharjan, S., Lysaker, O.M., Vaagsaether, K., Bjerketvedt, D.: Velocity and pressure along detonation fronts - Image processing of experimental results. In: Proc. of the Eighth International Seminar on Fire and Explosion Hazards (ISFEH8), pp. 133–149. (2016)
11. Law, C.K.: Combustion Physics. Cambridge University Press, New York, USA, (2010)
12. Settle, G.H.,: Schlieren and Shadowgraph Techniques. Springer-Verlag Berlin Heidelberg, New York.(2001)
13. Liepmann, H.W., Roshko, A.: Elemnents of Gasdynamics. Dover Publications Inc, New York. (2014)
14. Settle, G.S., Hargather, M.H.: A review of recent developments in schlieren and shadowgraph techniques. Measurement Science and Technology, 28. (2017).
15. Maharjan, S., Bjerketvedt, D., Lysaker, O.M.: Study of a Reflected Shock Wave Boundary Layer Interactions Based on Image Processing. Blind review process.

Acknowledgment

The experiments processed in this paper were conducted at California Institute of Technology (Caltech). Author would like to thank Prof. J.E. Shepherd and Dr. L. Boeck form Caltech and Prof. Ola Marius Lysaker from USN for their valuable contribution.

Article IV: Processing of High Speed Videos of Shock Wave Boundary Layer Interactions.

Processing of High Speed Videos of Shock Wave Boundary Layer Interactions.
Samee Maharjan, Dag Bjerketvedt and Ola Marius Lysaker.
Production Process, Signal, Image and Video Processing, (2020)

Signal, Image and Video Processing
https://doi.org/10.1007/s11760-020-01782-5

ORIGINAL PAPER

Processing of high-speed videos of shock wave boundary layer interactions

Samee Maharjan[1] · Dag Bjerketvedt[1] · Ola Marius Lysaker[1]

Received: 26 June 2019 / Revised: 16 March 2020 / Accepted: 7 September 2020

Abstract

This paper presents a framework for processing high-speed videos recorded during gas experiments in a shock tube. The main objective is to study boundary layer interactions of reflected shock waves in an automated way, based on image processing. The shock wave propagation was recorded at a frame rate of 500,000 frames per second with a Kirana high-speed camera. Each high-speed video consists of 180 frames, with image size [768 × 924] pixels. An image processing framework was designed to track the wave front in each image and thereby estimate: (a) the shock position; (b) position of triple point; and (c) shock angle. The estimated shock position and shock angle were then used as input for calculating the pressure exerted by the shock. To validate our results, the calculated pressure was compared with recordings from pressure transducers. With the proposed framework, we were able to identify and study shock wave properties that occurred within less than 300 µsec and to track evolveness over a distance of 100 mm. Our findings show that processing of high-speed videos can enrich, and give detailed insight, to the observations in the shock experiments.

Keywords Image processing · Front tracking · Shock wave · High-speed videos

1 Introduction

The introduction of charge-coupled device (CCD) and complementary metal-oxide semiconductor (CMOS) technology revolutionized high-speed photography during 1980s and 1990s. Today, there are numerous types of high-speed cameras that operate at a frame rate of more than a million images per second with resolution of one mega pixel. This development enables researchers to capture fast phenomena like the shock wave propagation and gas explosion in a video as a series of images [1]. Based on the advanced image and computer technology, these images now give an alternative source to estimate the shock wave characteristics like shock speed, pressure etc. [2,3]. However, in early days, images were mainly used for visualization of the phenomena [4]. From an image processing perspective, the ability to extract the desired information automatically was limited.

Image processing is an interdisciplinary research field, where the aim is to extract some desired information from images. It is widely used for object detection, object location, classification, segmentation and motion detection, among others. Based on improved computer power, sensor development and algorithmic progress, the applications have been broaden into various other fields, for example security system, road safety, document enhancing and gas dynamics. The study of a shock wave generally called shock on the other hand is one of the important research area in gas dynamics [5]. Over the years, the methods for estimating shock characteristics have evolved from local pressure sensor recordings and numerical simulations [4,6] to advanced simulation techniques [7]. The use of image processing in this particular field was evolved during early 2000s. One of the main advantages of using high-speed video/images over a traditional sensor-based approach is the possibility of extracting large amount of information within a small-scale experiment (< 100 mm). It makes it possible to extract information with intervals < 1 mm, meaning that several images may be recorded in between two pressure transducers. In [2,3], a rather simple image processing method using intensity difference successfully determined the shock front position in the images. These images were captured at low frame rate; thus, they are of a high quality. However, in case of higher frame rates, the quality of images degrades. Thus, improved image process-

✉ Samee Maharjan
samee.maharjan@usn.no

[1] University of South-Eastern Norway, Kjølnes ring 56, 3918 Porsgrunn, Norway

Published online: 28 September 2020

ing methods are needed. Some previously developed image processing framework to process similar high-speed videos and determining shock front position can be read in [8,9]. The filtering methods suggested in [8,9] have a multistep spatial filtering unit which increases the computing time. Furthermore, in some of the noiser images, the corresponding segmented images contain more background noise.

In this paper, we propose an image processing framework based on a fast preprocessing unit in combination with a robust tracking algorithm. The proposed framework processes images from high-speed videos, recorded during shock wave boundary layer interaction (SWBLI) experiments. The tracked wave fronts were further analyzed to estimate (a) the shock position; (b) position of triple point; and (c) shock angle. The pressure exerted by the reflected shock was calculated by using the incident shock speed, the shock angle and shock polare. For validation, the estimated pressure was compared with the pressure measurement from the transducer.

The rest of the paper is organized as follows. A description of the experimental setup and the high-speed videos is described in Sect. 2. The designed image processing framework along with the method to approximate the shock wave characteristics is described in Sect. 3. The estimated results are presented and discussed in Sect. 4 followed by the conclusions in Sect. 5.

2 Materials

2.1 Experimental setup

The experiments were conducted in the Graduate Aeronautical Laboratories at California Institute of Technology (GALCIT) detonation tube [2]. The tube is a 7.6 m long, 280 mm inner-diameter and equipped with a 152.4 mm wide test section and two polycarbonate windows to provide optical access.. The optical window constructed in the test section makes it able to capture shock wave boundary layer interaction when the incident shock is reflected back by the reflecting wall. For details of experimental setup, refer to [2]. The planar incident shock in the CO_2 gas was driven by a detonating slugg of $C_2H_2 - O_2$. The initial pressure was 13 kPa, and Mach number of the incident shock was 2.4.

2.2 Shock wave boundary layer interaction

A schematic diagram of a test section is illustrated in Fig. 1. As shown in Fig. 1a, the first event is a planar incident shock propagating toward the reflecting end wall. A boundary layer is simultaneously formed at the tube wall behind the propagating incident shock as seen in Fig. 1b. When this shock hits the reflecting end wall, the reflected shock then propagates backward illustrated in Fig. 1c and interacts with the boundary layer. A Mach stem structure with a triple point is formed as a result of the interaction which is shown in Fig. 1d. The phenomena is known as a bifurcation of a reflected shock wave. Mark reported a classic study of boundary layer interaction of reflected shocks in a shock tube in 1958 [4]. The distorted reflected shock close to the wall preceding the normal reflected shock is commonly called an oblique shock or the foot. The third shock behind these two shocks is known as rear shock. The point where all three shocks meet is what called a triple point, and the angle that the oblique shock makes with the boundary is a shock angle. The triple point configuration is also known as a Mach stem structure; further description of bifurcation of reflected shock and Mach stem can be found in [4,5].

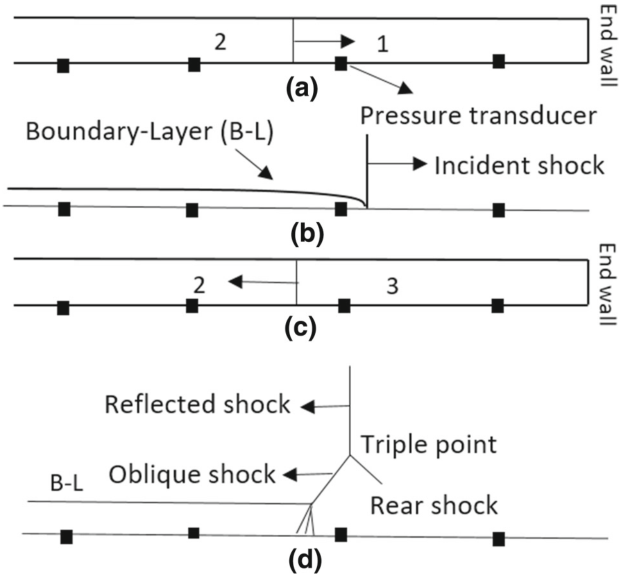


Fig. 1 Schematic representation of the shock wave propagating in a shock tube; **a** shock tube with incident shock; **b** formation of a boundary layer (B-L) behind the incident shock at the tube wall; **c** shock tube after the incident shock reflected off the end wall; **d** reflected shock interacts with the (B-L) and formed a triple shock configuration at the tube wall

2.3 Imaging technology

The two imaging techniques used for capturing the wave propagation, namely shadowgraph and schlieren [10]. They translate the phase speed difference of a light passing through the medium, into different intensities in a viewing plane (image). The use of these imaging technologies in the flow visualization can be read in [11]. For the review of recent developments in shadowgraph and schlieren techniques, refer to [1].

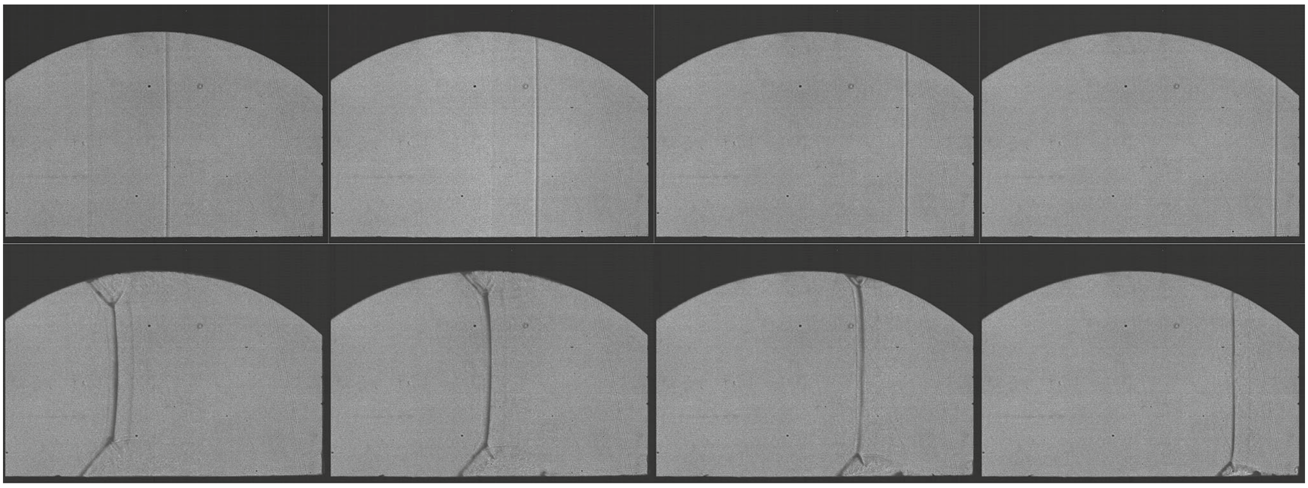


Fig. 2 Shadowgraph images showing the propagation of the incident shock (first row) and the reflected shock (second row) during Exp. 2516. Images are placed chronologically left-to-right in the first row and right-to-left in second row

2.4 High-speed videos

A high-speed camera especially introduced for capturing a fast phenomena like the compressible gas flow named Kirana [1] was used to record the SWBLI experiments. Even though the camera was operated at the frame rate of 500,000 frames per second, the SWBLI is an extremely fast phenomenon, which occurs in less than a millisecond. Thus each high-speed video consists of 180 images of size [768 × 924] pixels which is about [81 × 97] mm or [60 × 72] mm, depending on the scaling factor of 9.528 or 12.902 pixels/mm, respectively. In this paper, we processed two experiments, one done with schlieren technique (Exp. 2558) and another with shadowgraph (Exp. 2516). A sequence of images from the high-speed video (Exp. 2516) is shown in Fig. 2. The images in the first row show the propagation of the incident shock from left to right toward the reflecting end, and the images in the second row show the propagation of the reflected shock from right to left. The formation of a triple point structure is clearly visible in the second row.

3 Image processing

In short, an image processing framework was designed to perform these following tasks:

1. Read each image from a high-speed video.
2. Reduce background noise while preserving desired information.

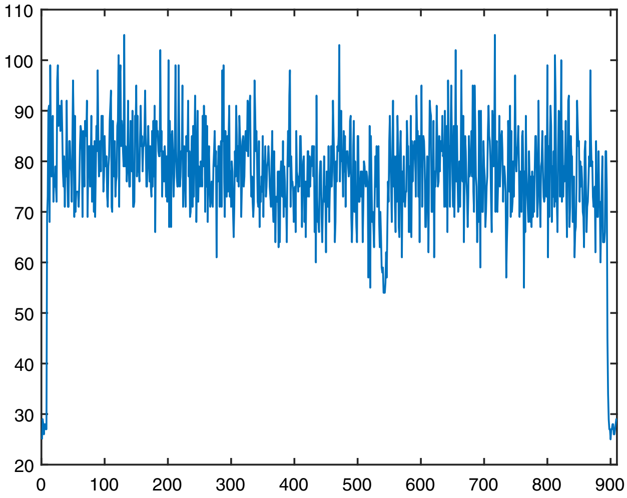


Fig. 3 Plot of the intensity values of a single row in one of the images from the high-speed video (Exp. 2516)

3. Track/locate the front of the shock wave in images with a visual front.
4. In case of reflected shock, segment the tracked front into a normal shock and an oblique shock.
5. Calculate the shock angle and the height of the triple point.

Due to the presence of the camera window in the upper part in all images (see Fig. 2), only the lower 400 rows were considered for the rest of this paper.

For the human eyes, the fronts are easily captured in the images, and however, to get a computer program to perform the same task is not trivial. To elaborate this, a 1-D intensity profile of a single row is given in Fig. 3. This figure shows the ambiguity of the solution. The framework to process high-

[1] https://www.specialised-imaging.com/products/video-cameras/kirana.

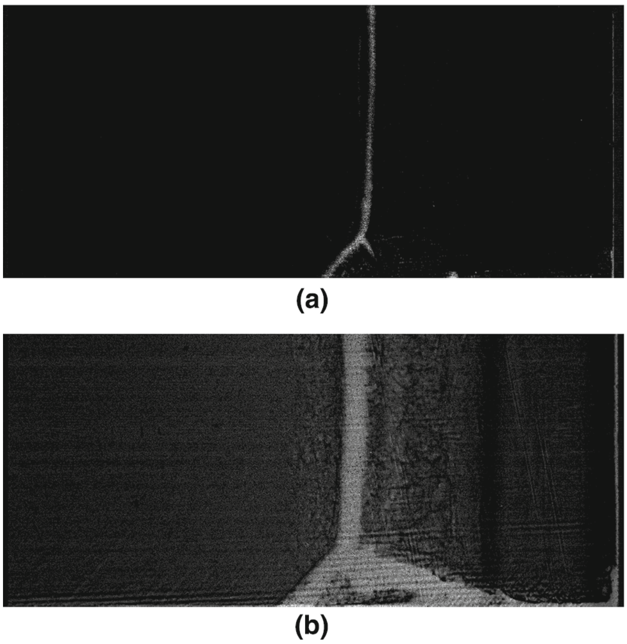


Fig. 4 Examples of background subtraction when applied to **a** bottom left image in Fig. 2; **b** one of the image from Exp. 2558

speed videos and to overcome the above-mentioned problem is described in the following subsections.

3.1 Image filtering

Background noise in the images comes mainly from the filming technology, i.e., the camera, the experiment equipments and ongoing changes in gas properties. The filtering of an image in the current framework consists of a background subtraction followed by a low-pass frequency filtering. For background subtraction, a background image was generated from each high-speed video by averaging all images without a front. The number of images without fronts varies within each video, and sometimes there might be several images and sometimes just one. The background image was constructed with a prior study of the images and manually choosing the images without shock fronts. The constructed background image was then successively subtracted from all the images with a visual front. The result of background subtraction in one of the images from each experiment is shown in Fig. 4.

Filtering of an image can be defined as an operation in which the value of the any output pixel is determined by the combination of the values of the pixels in the neighborhood of the corresponding input pixel. Convolution is a filtering algorithm which takes a weighted sum of the neighboring input pixels, and the matrix defining the weights is known as convolutional kernel. Filtering using the convolution of a kernel matrix with the original image in the spatial domain is operationally costly, when the image size is too large. In such cases, filtering can be done in frequency domain where the multiplication is identical to the convolution in spatial domain.

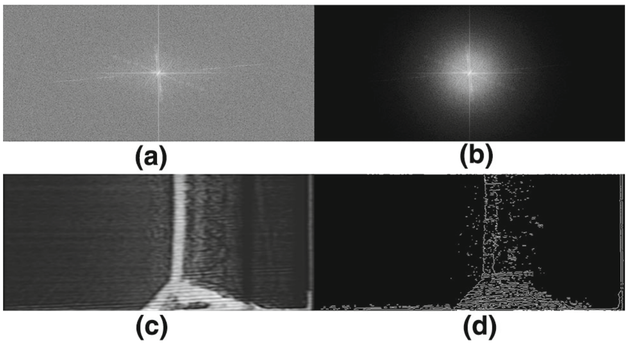


Fig. 5 Low-pass frequency filtering corresponding to Fig. 4b; **a** frequency spectrum; **b** frequency spectrum of filtered image; **c** filtered image in spatial domain after IFFT; **d** result of edge detection ’prewitt’ corresponding to Fig. 5c

For filtering in frequency domain, at first the original image (spatial) was transferred into frequency domain by using fast Fourier transform (FFT). Then, multiplied the transferred image with the filter function and thereafter, re-transformed back to the spatial domain by using inverse Fourier transform. A low-pass filter (LPF) attenuates high frequencies greater than a cutoff frequency, resulting a smoother image in the spatial domain. A low-pass filter of order 3, having a cutoff frequency of 50 Hz, was used for filtering the images. The frequency spectrum of Fig. 4b can be seen in Fig. 5a and Fig. 5b is the result after multiplication of Fig. 5a with LPF, whereas Fig. 5c is the final filtered image. Further details regarding fast Fourier transform and filtering of images in frequency domain can be read in [12]. Figure 5d shows the result of using ’prewitt’ edge detection method from MATLAB image processing toolbox corresponding to Fig. 5c. Some of the available edge detection methods in various processing toolboxes were able to detect the edges; however, they did not provide the required clarity of front position. This suggested further processing of the filtered images in order to track the exact front position.

3.2 Segmentation

The term image segmentation refers to the partition of an image into a set of regions that cover the entire image. For our application, this implies separating the shock front from the background. To do so, all images were normalized to intensity values in range [0–1]. The images were segmented into a background (black) and the shock wave (white) based on a thresholding technique suggested by Otsu [13]. The Otsu’s algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels). The algorithm then calculates the

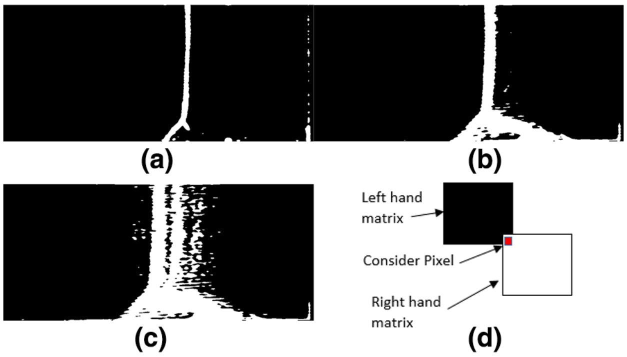


Fig. 6 The result of segmentation (after low-pass filtering) performed in; **a** Fig. 4a; **b** Fig. 4b; **c** an example of segmentation where filtering did not suppress all noise; **d** a template consists of two 5 × 5 matrices for the considered pixel

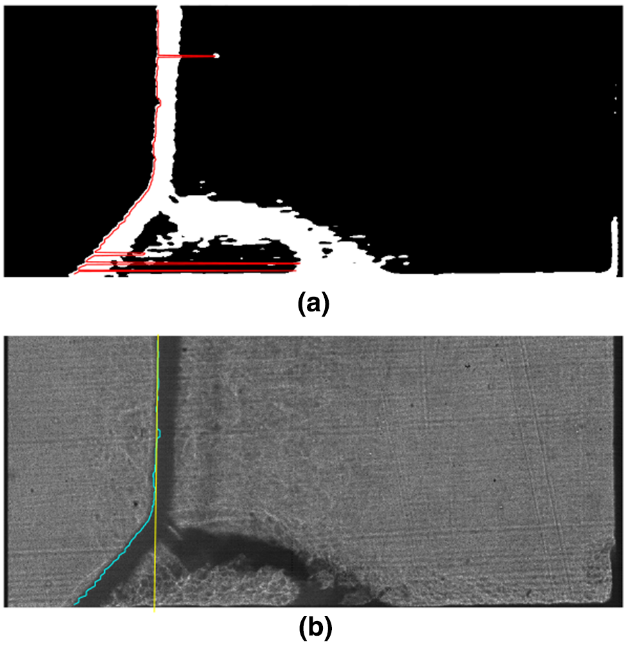


Fig. 7 **a** A segmented image from high-speed video of Exp. 2558 with the front tracked by template matching. The tracking misplaced some points in the lower part of the front; **b** second tracking done only till median of first tracked front(yellow line) shown in the corresponding raw image (color figure online)

optimum threshold separating the two classes with minimum variance. For example, a threshold value given by Otsu's method corresponding to Fig. 5c was 0.25. All the pixels with an intensity level below 0.25 are labeled as background, and all pixel values equal to or greater than 0.25 are labeled as foreground. The result of applying Otsu's algorithm (after low-pass filtering) in Fig. 4a and b is shown in Fig. 6a and b, respectively.

3.3 Front tracking using template matching

To get an algorithm to track the front in an image like Fig. 6a,b is rather simple, just slide through all pixels and mark the first white pixel (from left) in each row. However, the output from the segmentation step, i.e., Otsu's algorithm, does not always work that perfect. The amount of noise varies from image to image, video to video, and can affect the segmentation result shown in Fig. 6c. Thus, to overcome these situations, a template matching technique was designed for automatic track of the fronts. A template matching technique is a well-known method for image classification and segmentation. In short, with this approach, the aim is to find the position/object in the scene which match best with a predefined template [14]. For the present work, a template which consists of two 5 × 5 matrix is created as shown in Fig. 6d. As it can be seen in Fig. 6d, the matrices are shifted such that a single template can be used for tracking both the normal (straight) and oblique (tilted) shock. The element values for the left-hand matrix of the template are set to 0 (black), while for the right-hand matrix to be 1 (white). The framework matches the template pixel by pixel, row by row, and calculates a matching error for each location. The matching error is the sum of the differences between the pixel values of the template and the footprint template created around the considered pixel in the image. For example, in Fig. 6a–c, around the top left of the image where only background is present, the left-side matrix of the template will match perfectly. However, the right-side matrix will not match at this location, thus resulting a huge error. Moving further along the columns, the template will give minimum error around the front where both the matrices seem to match perfectly. To eliminate the loss of result due to boundary pixels in the lower boundary, the last row was replicated for five more rows than the actual data. After the matching process, the pixel in each row where the template matches the best, i.e., the point with the minimum error was chosen. For most of the images, the tracking was accurate; however, for a few images in Exp. 2558, some mistracked points were observed, as seen in Fig. 7a. To overcome this problem, the following a priori information was incorporated:

1. The oblique shock wave is ahead of the normal shock.
2. The position of triple point does not goes above row no. 150.

The median of the normal shock (row 50:150), tracked at the first (green curve in Fig. 7a) was calculated which is shown by the yellow line in Fig. 7b. Later, during the second tracking, the framework tracks the front only till the calculated median, such that the noise present beyond the median will be excluded. The median line with the final front is shown in Fig. 7b.

3.4 Estimating triple points based on segmented regression

To estimate the triple point and the shock angle, the tracked front was divided into a normal shock and an oblique shock, by using a segmented regression technique. Segmented regression is a method of fitting multiple lines from a single dataset [15]. In case of fitting two lines from a single dataset, it requires only one break point BP, and the model can be written as,

$$y = a_1x + b_1 \quad \text{for } x \leq \text{BP} \tag{1}$$

$$y = a_2x + b_2 \quad \text{for } x > \text{BP}. \tag{2}$$

In our case, the x-y points of the tracked front served as a dataset from which two straight lines were fitted: one for the normal shock and one for the oblique shock, as illustrated in Fig. 8. Please note that, the segmented regression process was conducted in the x-y coordinate system. Figure 9 summarizes the segmented regression process of determining the optimum BP by consecutively fitting two lines to the underlying dataset. For the sake of understanding, the process is demonstrated in the images. The blue curve represents the tracked front, and the yellow mark yields the breaking point BP, while the orange line represents the first line fitted as in (1). The white line is then fitted to the remaining points below BP as in (2). The process follows a brute force approach, from top to bottom, starting BP at row no. 10 and for each iteration, the BP moves down by one row. After each iteration, the least square error is calculated for both fitted lines with their respective location and the errors are summed up and stored. For example, in Fig. 9a the orange line will give a small amount of error as it almost coincides with blue curve above the yellow point, while the white line is misplaced and causes large errors. Figure 9b shows the fitted line when the separating point is located near the triple point and thereby the fitted lines representing the normal and the oblique shock are rather accurate. The total error gathered from both lines seems to be at a minimum with this location. Further down, see Fig. 9c, the error increases. For each image, after the process was finished, the BP that gave the minimum error was considered as the separating point between the normal and the oblique shock. A close look around the triple point is presented in Fig. 9d, where the point which gives the minimum error between two fitted lines is visualized with a yellow mark. Now, based on the selected point (yellow), two straight lines are refitted, the red line with respect to points from the tracked front below the yellow point, whereas the green one is fitted with respect to 100 points above it. (By trial and error 100 points were used in this study, but other selections may be used). The white circular point shows the crossover between the two fitted lines, i.e., the triple point, while the slope of the red line gives the shock angle.

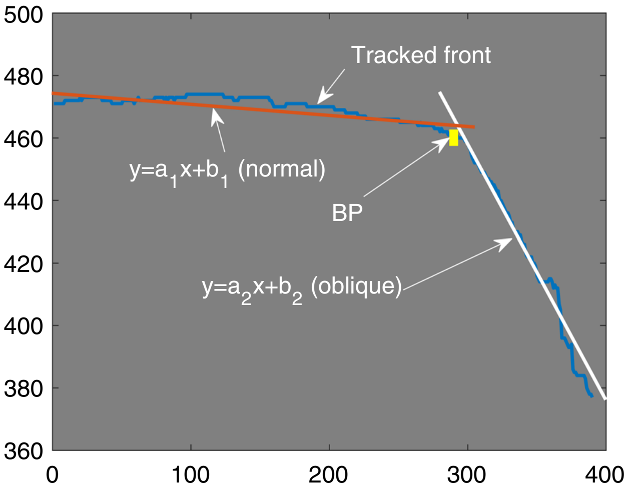


Fig. 8 The position/height of a BP with respect to the lower boundary in x-y coordinate system

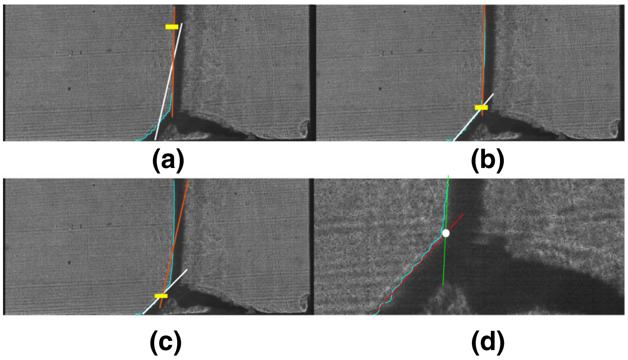


Fig. 9 **a–c** Segmented regression process. The yellow marks represent different separating points for line fitting. The blue curve represents the tracked front. The orange line yields a line fitted for the normal shock, while the white line gives the line fitted for the oblique shock; **d** a close look at the triple point area (color figure online)

4 Results and discussion

In this work, an image processing framework including frequency filtering, template matching and segmented regression was developed to track the wave front in multiple high-speed videos from shock experiments. The fronts tracked in Exp. 2516 are presented in Fig. 10. Figure 11 shows the comparison between the proposed algorithm with previous work [8,9]. The segmented images in Fig 11a and b demonstrate that the segmentation with FFT filtering performs better than median filtering used in the segmented images demonstrates that the segmentation with FFT filtering performs better than median filtering used in [8,9]. In addition, the average time taken by the proposed method to track the front is about 4 sec per image, which is a second less than by [9] and around 20 sec (45 iterations) less than [8]. It can be observed in Fig 11d that the front tracked by the proposed method is closest to

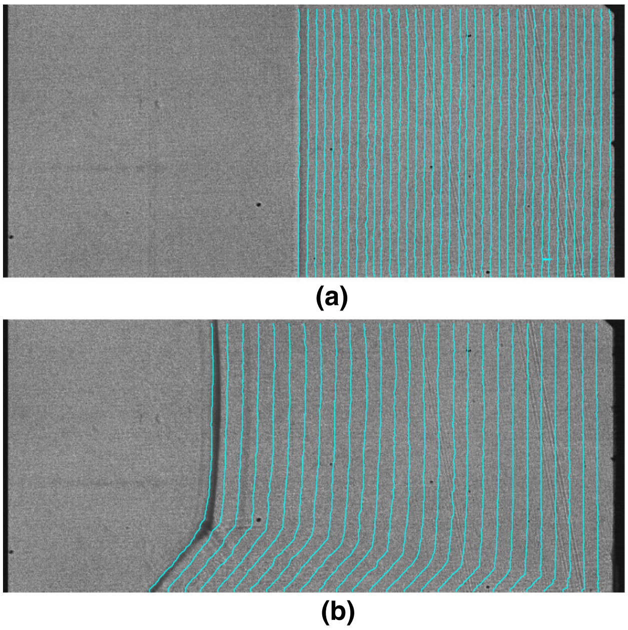


Fig. 10 Collection of the tracked fronts in the Exp. 2516; **a** incident shock (fronts plotted in each time step); **b** reflected shock (fronts plotted in every five time step)

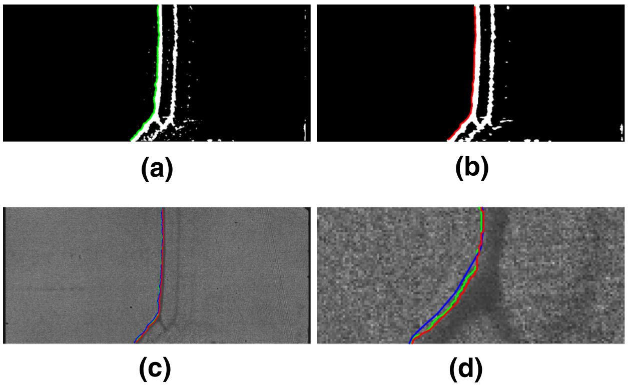


Fig. 11 Comparison between results from proposed framework with results from [8,9]; **a** the segmented image and a tracked front by [9]; **b** the segmented image and the front tracked by proposed method; **c** the fronts tracked by all three methods (blue— [8], green—[9] and red—proposed method; **d** enlarged around triple point (color figure online)

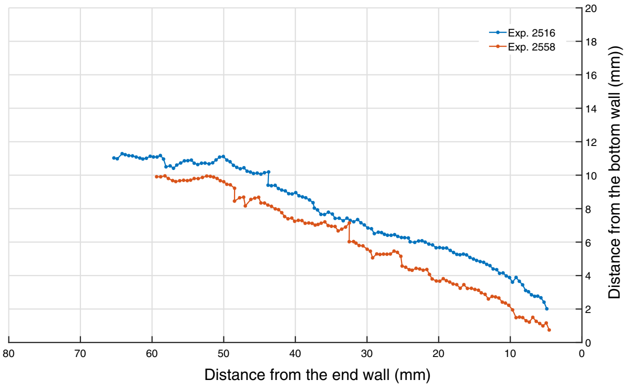


Fig. 12 The estimated height of the triple point from the lower boundary, orange marks for Exp. 2516 and blue marks for Exp. 2558 (color figure online)

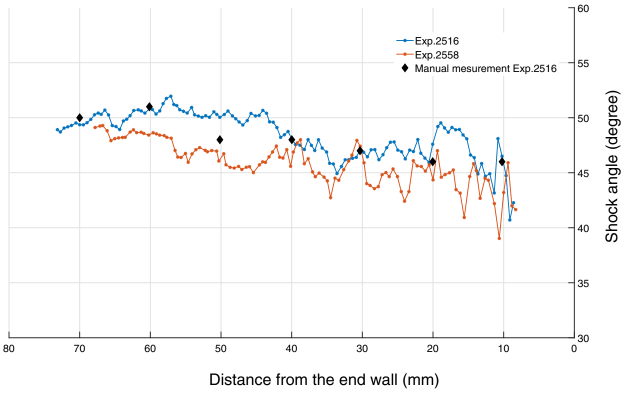


Fig. 13 The estimated shock angles, orange marks for Exp. 2516 and blue marks for Exp. 2558 (color figure online)

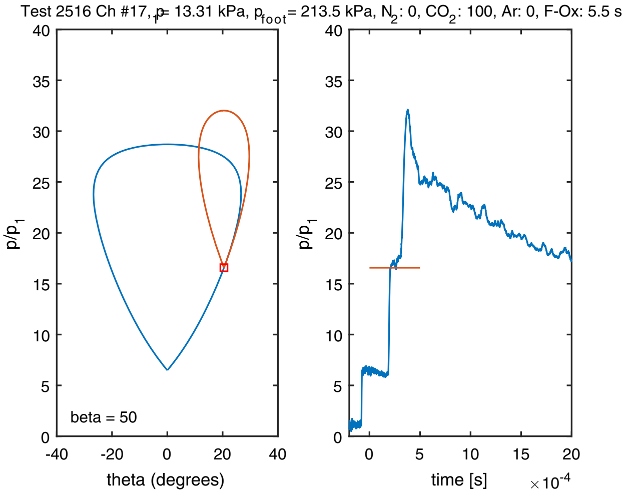


Fig. 14 A shock polar plotted for Exp. 2516 with shock angle 50 degree, and the pressure reading in a pressure transducer (channel 17)

the actual front. However, the differences are small and is a subjective matter.

The first few frames are excluded while calculating presenting the heights of triple points and shock angles as the number of points representing the oblique shock was too sparse to have a valid calculation. The heights of the triple points from the lower wall have a likewise behavior in both experiments as can be observed in Fig. 12. The height increases as the front moves away from the reflecting wall, however at the later part beyond 50 mm from reflecting wall, and the height flattens out. A similar behavior has earlier been observed by [16]. The shock angles calculated for both experiments are plotted in Fig. 13. The black diamond shape marks in Fig. 13 represent the manual measurement performed on the images of Exp. 2516. The images were selected such that the position of foot of shock lies around 10,20,30…mm from the reflecting wall. The comparison with the manual measurements shows that the shock angle calculations are in good agreement.

Further, to compare the results derived from the image processing with actual pressure recording, a graphic method known as the shock polare [11] was used. Basically, it makes possible to find the shock strength, i.e., pressure ratio across the shocks (both normal and oblique), from knowing the Mach number (speed) of the incident shock and the angle of the oblique shock. The incident shock speed was calculated by using nonlinear regression for line fitting on the position of the incident shock [2]. The shock polare for Exp. 2516 is shown in Fig. 14 (left). The blue curve in Fig. 14 (left), gives the relation between deflection angle (theta) and the pressure ratio across the oblique shock (p/p1). When the shock angle, as shown in Figs. 2 and 13, is known, the deflection angle can be calculated with shock equation [11].

Figure 14 (right) shows the pressure recordings of the transducer located 50 mm from the reflecting wall. We manually select the image from the high-speed video in which the foot position of the wave was located closest to the selected transducer. The corresponding shock strength in the shock polare agrees with the shock strength recorded in the pressure recording as shown by red marks. Another shock polare (red curve) originated at shock strength point in the blue curve is the shock polare for the rear shock (see Fig. 2). Behind the normal shock and the rear shock, deflection angle and pressure will be the same [11]. Therefore, the upper crossing of the blue and the red polare represents this solution for these parameters.

So, from image processing we are able to predict the pressure and the state in the triple point configuration (above the boundary layer). This information will not be available from a pressure recordings, since pressure will be influenced by boundary layer interactions as they were mounted in the wall. To this end, it can be concluded that the high-speed videos in combination with image processing can enrich and give new detailed insight to shock wave boundary layer interactions.

5 Conclusions

In this paper we have used high-speed video in combination with image processing as a framework for studying reflected shock wave and shock wave boundary layer interactions in detail, within time intervals of 300 μs. Frequency filtering showed an overall accurate performance both with respect to robustness and precision for the application at hand. Template matching is a simple, but still powerful technique for identifying specified features in an image. In case of front tracking, misplacements of the front was seldom observed in this study. Segmented regression proved to be an efficient tool for dividing the tracked wave front into two parts: the normal and the oblique shock. However, the template matching and the segmented regression took almost 80 percent of total processing time. Hence, further work can be done to obtain a speed-up concerning these steps. The pressure ratio across the shock wave was calculated with a shock polare, which then was compared with the measurements from actual pressure transducers. The comparison revealed that our estimates are in good agreement with the pressure recordings. The overall processing of one high-speed video takes less than an two hour to complete which is comparatively lesser than any small computer simulation done for the same study. With the proposed framework, we were able to identify and study shock wave properties that occurred within less than 300 μsec and to track evolvement over a distance of 100 mm.

Acknowledgements The experiments processed in this paper were conducted at California Institute of Technology (Caltech). Authors would like to thank Prof. J.E. Shepherd and Dr. L. Boeck for their valuable contribution.

Funding Open Access funding provided by University Of South-Eastern Norway.

References

1. Settle, G.S., Hargather, M.H.: A review of recent developments in schlieren and shadowgraph techniques. Meas. Sci. Technol. **28**, 042001 (2017). https://doi.org/10.1088/1361-6501/AA5748
2. Damazo, J.S.: Planar Reflection of Gaseous Detonations. Doctoral Thesis, California Institute of Technology, Pasadena, California (2013)
3. Timmerman, B.H., Skeen, A.J., Bryanston-Cross, P.J., Tucker, P.G., Jefferson-Loveday, R.J., Paduano, J.D., Guenette, G.R.: High-speed digital visualization and high-frequency automated shock tracking in supersonic lows. Opt. Eng. **48**, 10 (2008). https://doi.org/10.1117/1.2992621
4. Mark, H.: The Interaction of a Reflected Shock Wave With the Boundary layer in a shock tube. Cornell University, Ithaca, New York (1958)
5. Babinsky, H., Harvey, J.K.: Shock Wave-Boundary-Layer Interactions. Cambridge University Press, London (2012). https://doi.org/10.1017/CBO9780511842757
6. Kleine, H., Lyakhov, V.N., Gvozdeva, L.G., Grönig, H.: Bifurcation of a reflected shock wave in a shock tube. In: Takayama, K. (ed.)

Shock Waves, pp. 261–266. Springer, Berlin, Heidelberg (1992). https://doi.org/10.1007/978-3-642-77648-9-36

7. Kryukov, A., Ivanov, I.E.: Shock wave-boundary layer interaction in a long shock tube. J. Phys. Conf. Series **1009** (2017). https://doi.org/10.1088/1742-6596/1009/1/012010
8. Maharjan, S., Gaathaug, A.V., Lysaker, O.M.: Open active contour model for front tracking Of detonation waves. In: Proceedings of the 58th Conference on Simulation and Modelling, pp. 174–179. Linkoping University Electronic Press, Sweden (2017). https://doi.org/10.3384/ecp17138174
9. Maharjan, S., Bjerketvedt, D., Lysaker, O.M.: An image processing framework for automatic tracking of wave fronts and estimation of wave front velocity for a Gas experiment. In: Representation, analysis and recognition of shape and motion From Image data. Springer International Publishing, p. 842 (2019). https://doi.org/10.1007/978-3-030-19816-9
10. Settle, G.S.: Schlieren and Shadowgraph Techniques. Springer-Verlag, Berlin (2001). https://doi.org/10.1007/978-3-642-56640-0
11. Liepmann, H.W., Roshko, A.: Elements of Gasdynamics. Dover Publications Inc, New York (2014)
12. Najim, M.: Digital Filters Design for Signal and Image Processing. Wiley Press, London, UK (2010)
13. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076
14. Dufour, R.M., Miller, E.L., Galatsanos, N.P.: Template matching based object recognition with unknown geometric parameters. IEEE Trans. Image Process. **11**(12), 1385–1396 (2002). https://doi.org/10.1109/TIP.2002.806245
15. Ryan, S.E., Laurie, S.P. : A tutorial on the piecewise regression approach applied to bedload transport data. General technical report RMRS-GTR-189. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station., (2007). https://doi.org/10.2737/RMRS-GTR-189
16. Taylor, J.R., Hornung, H.G.: Real gas and wall roughness effects on the bifurcation of the shock reflected from the end wall of a tube. Shock tubes and waves. In: Proceedings of the 13th International Symposium, pp. 262–270 (1982)

Article V: Information Extraction from High Speed Videos of Reflected Shock Wave Interaction With Boundary Layer.

Not available in the electronic version of the dissertation

Information Extraction from High Speed Videos of Reflected Shock Wave Interaction With Boundary Layer.

Samee Maharjan, Dag Bjerketvedt and Ola Marius Lysaker.

Well prepared and will submit to Shock Waves.

Part III

Unpublished work

Machine learning

Introduction

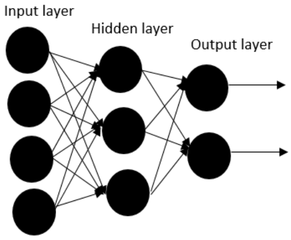


Figure 6.1: An example of DNN architecture.

If there is one field that has developed leaps and bound in the past decade, it should be Machine Learning. Machine learning essentially deals with the problem of automatically finding a decision, for example, separating cats from dogs, but by training the machine. During a training phase, the so-called training data set is pre-processed and meaningful features are extracted (Maier et al., 2018). In this context, one way of achieving is by implementing a neural network. The neural network consists of interconnected neurons that takes input and perform some processing on the input data, and finally forward the current layer output to the coming layer. Each neuron in the network sums up the input data and applies the activation function to the summed data and finally provides the output that might be propagated to the next layer. A single neuron itself can already be interpreted as a classifier if the activation function is chosen such that it is monotonic, bounded, and continuous for example sign function, sigmoid function (Cybenko, 1989).

A neural network that consists of multiple hidden layers is known as Deep Neural network (Schmidhuber, 2015). In the literature, many arguments are found why a deep structure has benefits for feature representation (Bengio et al., 2013). The general architecture of DNN is shown in Figure 6.1.

CNNs purposed by Farabet et al. (2013) has become the leading architecture for most image recognition, classification, and detection tasks from the early 90s. CNN is a type of deep learning neural network which is specially designed for processing the data in the form of images. A general structure of CNN is presented in Figure 6.2. CNNs differs from the usual neural network as it consists of a convolution layer which only consider a local

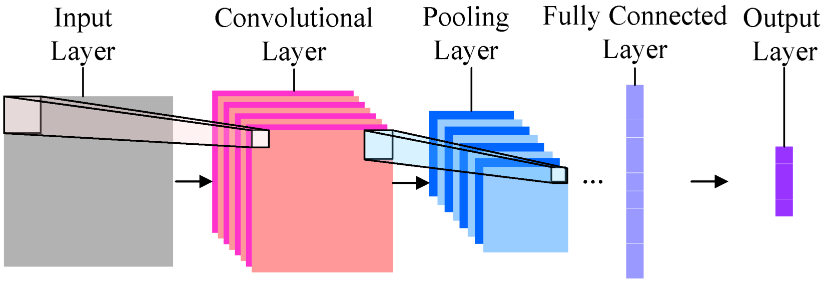


Figure 6.2: An example of CNN architecture [1].

neighbourhood for each neuron and that all neurons of the same layer share the same weights, which dramatically reduces the number of parameters and therefore memory required to store such a layer. Convolution is usually followed by pooling to reduce the scale of the input. The end of the network consists of a fully connected layer for output and the layer that defines the job of the network for example classification, regression, etc. CNNs have obtained great successes for various applications in computer vision, such as image classification (Krizhevsky et al., 2012), object detection (Girshick et al., 2014), image labelling (Farabet et al., 2013).

The possibilities of implementing machine learning to track the wave front in the high speed videos and subsequently calculating the primary and secondary information were studied. Due to time bound the algorithm cannot be completed. However, a simple 1-D deep neural network (DNN) and a 2-D convolutional neural network (CNN) model was designed.

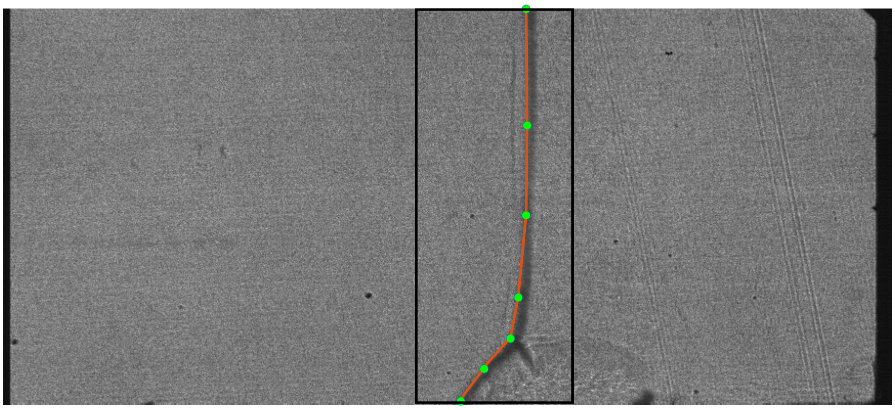


Figure 6.3: Manually plotting the front positions by choosing some points.

[1] https://www.frontiersin.org/articles/10.3389/fpsyg.2017.01745/full.

Training dataset

One of the high speed videos was selected to begin with and to reduce complexity only images with the reflected shock wave were considered. From the images with reflected shock, 14 images (10 percent), ranging from first to last every 10 frames apart was selected and the fronts were plotted manually. The fronts were plotted by picking a few points from top to bottom of the image, as shown in Figure 6.3 by yellow dots and a red line connecting them. By using extrapolation on the top boundary, it was made sure that the size of the front is always [400 × 1] , so that each row in the image have the front value. To generate more images for training, 56 training images generated from the original 14 by adding random noise.

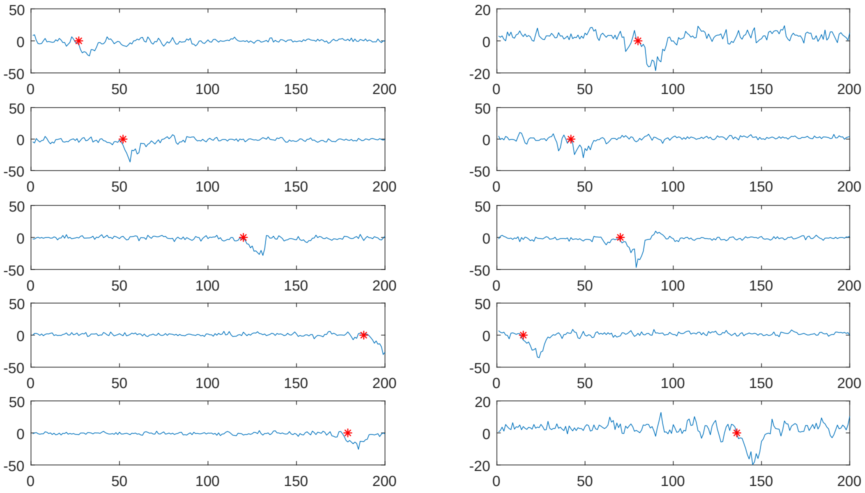


Figure 6.4: Some of the training data for DNN.

1-D dataset for DNN:

Due to the large size of the image [400 × 924], the images were cropped into [400 × 200] as shown by the cropping window in Figure 6.3 by a black rectangle. To make these 2-D images into a 1-D dataset, each row in training images was considered as one input and the position of the front in that row as an output. Such that an input is a vector of grayscale value of a row i.e. [1 × 200], and the output is the column number where the front is located. In total from 56 training images provides 22400 pairs for training. Among these randomly selected 10000 pairs were used as a training data set and 1200 for validation. Some of the examples of 1-D input data are shown in Figure 6.4 where the output is pointed by a red dot. After many combinations, the efficient one was a DNN with three layers each with 50 neurons and trained with 5000 epochs.

2-D dataset for CNN:

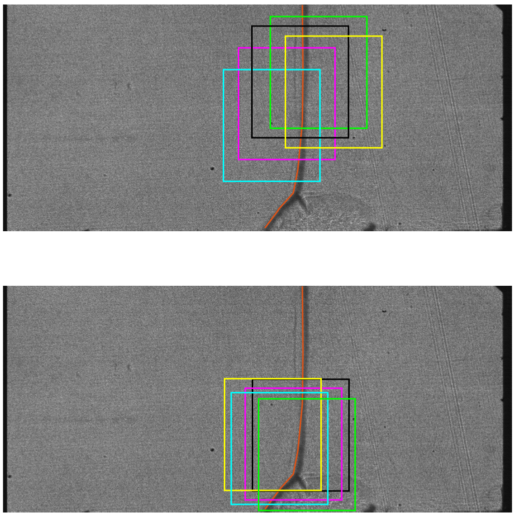


Figure 6.5: Generating multiple training images by sliding cropping window of size [200 × 200].

The most challenging part with CNN was creating a training database. Unlike some other types of images like numbers, real objects like cats and dogs, there was no training dataset available. To make image size trainable, we decided to work with an image size of [200 × 200], such that an entire oblique shock which spreads around 200 columns can be trained as well. A training set of 10000 images was made by cropping each training image into multiple [200 × 200] images by moving the cropping window left right and down as shown in Figure 6.5. Some of the training data are presented in Figure 6.6 and Figure 6.7. To reduce complexity, two CNN each for normal and oblique shock were trained. The best and efficient network architecture consists of an input layer at the beginning, three sets of convolution layer followed by normalization layer, relu layer, and max pooling layer and fully connected layer at the end. All the convolutional layers contained a filter of size [10 × 10] and the number of the filter was 8,16 and 32 respectively. The pooling layer is with a pool size of [2 × 2].

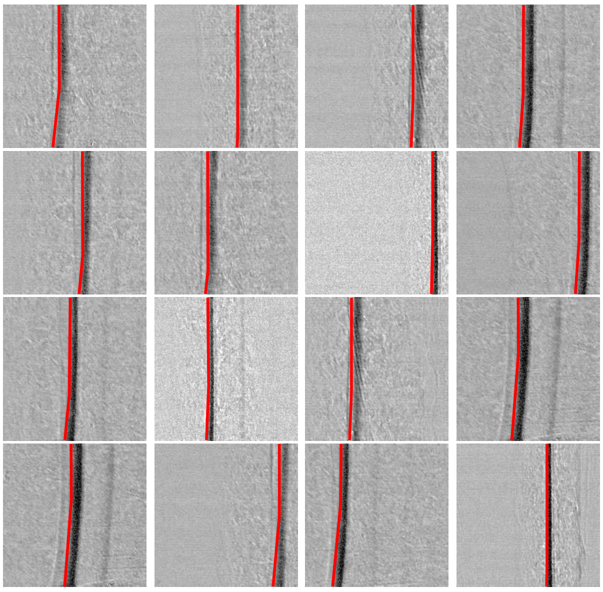


Figure 6.6: Some of the normal shock training data for CNN.

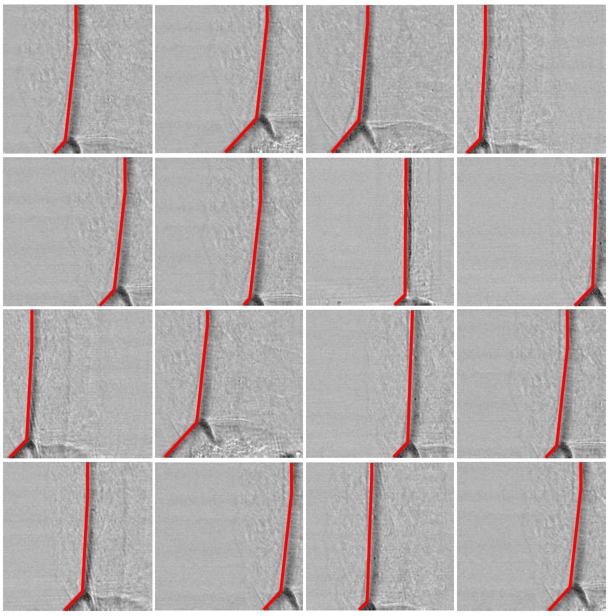


Figure 6.7: Some of the oblique shock training data for CNN.

Results

The mean square error for the validation of 1200 data was 2.2432. The result of the DNN on a few of the images excluding 14 training images, is shown in Figure 6.8. As it can be

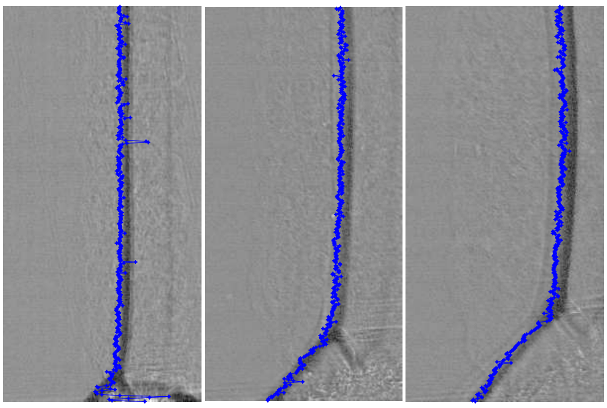


Figure 6.8: The result of DNN in three images (excluding training images) from the same training high speed video.

observed, there are some errors in the first image around the small oblique shock. The network tracks fairly better in later images. Considering the simple network, the output was satisfactory. The same network was used for tracking fronts in other high speed videos, an example is shown in Figure 6.9. The network tracks properly in the straight shock, however it was not able to track oblique shock. The initial reason assumed is the difference between the oblique shock structure in two high speed videos.

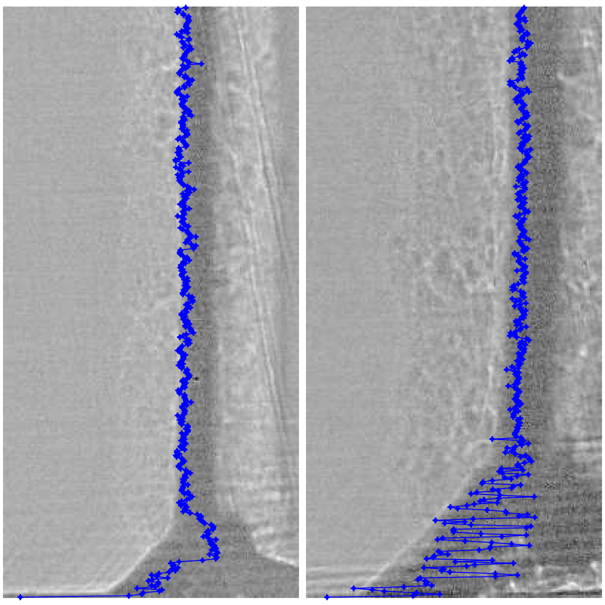


Figure 6.9: The result of DNN in the images from other high speed video.

Figure 6.10 shows the test results from CNN in some of the images from the training high speed video. The fronts in the upper halves are tracked by network trained with normal shock and the lower halves by network trained with oblique shock. As can be seen in the lower boundary, even in the first pic where the oblique shock is very small the tracking is good. However, there is some misplacement in normal shock.

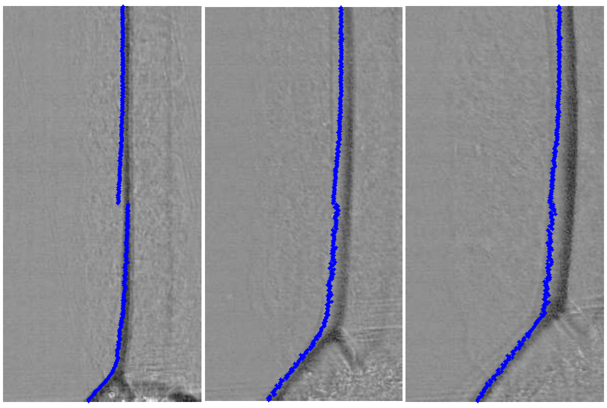


Figure 6.10: The result of CNN in three images (excluding training images) from the same training high speed video.

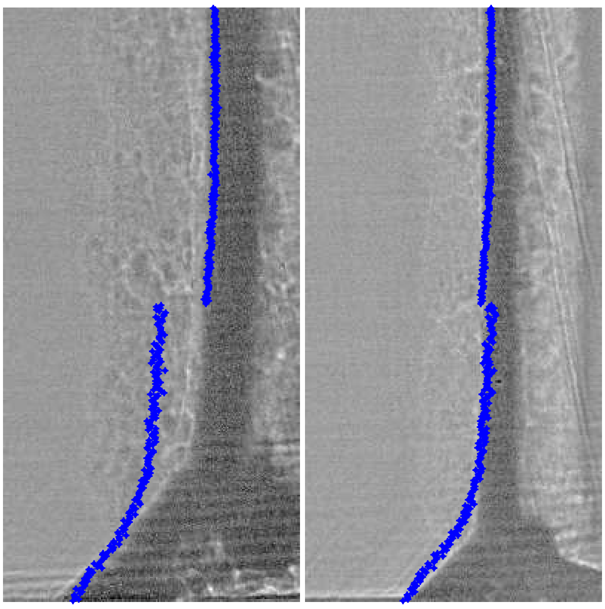


Figure 6.11: The result of CNN in the images from other high speed video.

Similar to DNN, the trained network shows some misplacement of oblique shock, while used in another high speed video as shown in Figure 6.11. At this time, the initial assumption for mis-tracking in normal shock is, the wave fronts used for training straight not curve however, the actual front is little curved.

Initial observation

1. A simple ANN works fairly good but only for the images from the training high speed video.
2. If the wave front is different from the training one, the accuracy of the network deteriorates massively.

Doctoral dissertation no. 83
2020

—

An Image Processing Framework for High Speed Videos from Combustion and Gas Explosion Experiments

Dissertation for the degree of Ph.D

—

Samee Maharjan

—

ISBN: 978-82-7206-579-8 (print)
ISBN: 978-82-7206-580-4 (online)

usn.no