

FMH606 Master's Thesis 2017  
Industrial IT and Automation

# **Wave Front Tracking using Template Matching and Segmented Regression**

Eirik Siljan

Faculty of Technology, Natural sciences and Maritime Sciences  
Campus Porsgrunn

**Course:** FMH606 Master's Thesis, 2017

**Title:** Wave Front Tracking using Template Matching and Segmented Regression

**Number of pages:** 49 report + 11 appendices = 60

**Keywords:** Wave front, tracking, segmented regression, template matching, triple point.

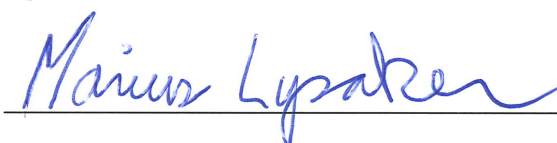
**Student:** Eirik Siljan

**Supervisor:** Marius Lysaker

**External partner:** Combustion, Process Safety and Explosions research group  
at USN with Dag Bjerketvedt

**Availability:** Open

**Approved for archiving:**  
(supervisor signature)



**Summary:**

The Combustion, Process Safety and Explosions research group at USN has collaborated with California Institute of Technology (Caltech) and has produced high-speed videos of shock wave fronts. These videos require a framework for wave front tracking through image processing.

The framework should calculate the characteristics of the wave front, i.e. the angle and velocity of normal and oblique shocks, as well as triple point location and velocity.

The framework was developed in MATLAB. It uses template matching and segmented regression to track the wave front in the high-speed videos.

An extended Segmented Regression variant is developed and used with two line segments and two separate datasets where the first segment is to be fitted with the front found by the normal shock template, and the second segment is to be fitted with the points found by the oblique shock template.

A "sliding window"-technique is implemented to reduce the search area of the templates. The user of the framework marks the wave front in the first and last frame. It is then possible to approximate how far the sliding window should move in each frame to follow the wave front, as well as the width of the sliding window.

Once the wave front has been tracked in all frames, the velocity is calculated based on the number of pixels the normal shock, the oblique shock and the triple point has moved in a known number of frames.

The result of the applied techniques is a framework which tracks a wave front from one frame to the next. In addition, the velocity and angle of the normal and oblique shock are presented, as well as the triple point velocity.

*The University College of Southeast Norway takes no responsibility for the results and conclusions in this student report.*

# Preface

This report is a Master Thesis written by a student of Industrial IT and Automation at University College of SouthEast-Norway (USN). It is written the 4<sup>th</sup> semester under the supervision of the Combustion, Process Safety and Explosions research group at USN. The task description is found as Appendix A.

The technique presented in this thesis is currently under evaluation for being published in SIMS (Scandinavian Simulation and Modelling Society). The initial abstract has been approved and can be seen in Appendix B. A full-length draft is currently under way for final approval.

Intermediate understanding of image processing is required to fully understand the report.

Almost all high-speed video images presented in this thesis has been brightened to enhance paper printed image quality.

The computer tools used in this project are:

- Calculations and development: MATLAB 2016b
- Report and attachments: Microsoft Word 2016
- Illustration: Visio 2016

Thanks to:

- Supervisor, Marius Lysaker
- PhD. student Samee Maharjan
- Combustion, Process Safety and Explosions research group at USN with Dag Bjerketvedt
- California Institute of Technology (Caltech)

Porsgrunn, 15.05.2017

Eirik Siljan

# Nomenclature

CPSE - Combustion, Process Safety and Explosions research group at USN

TP - Triple Point

NS - Normal Shock

OS - Oblique Shock

# Contents

Preface .....	3
Nomenclature .....	4
Contents .....	5
<b>1..Introduction .....</b>	<b>7</b>
1.1 Background .....	7
1.1.1 Test Rig .....	8
1.2 Task Description .....	9
1.3 Constraints and Extensions .....	10
<b>2..Literature Study .....</b>	<b>11</b>
2.1 Image Thresholding .....	11
2.2 Edge Detection .....	12
2.3 Template Matching .....	14
2.4 Watershed.....	15
2.5 Fourier Transform in Image Processing.....	18
<b>3..Methods .....</b>	<b>19</b>
3.1 Outlier Frames.....	19
3.2 Wave Front Template Matching.....	19
3.2.1 Normal Shock Template .....	19
3.2.2 Oblique Shock Template .....	21
3.2.3 Outlier Correction .....	23
3.3 Sliding Window .....	23
3.4 Segmented Regression: Normal and Oblique Shock .....	24
3.4.1 Segmented Regression .....	25
3.4.2 Extended Segmented Regression .....	25
3.5 Normal Shock Estimation .....	28
3.6 Oblique Shock: Secondary Tracking .....	29
3.7 Angle Estimation.....	31
3.8 Velocity Estimation.....	31
3.9 Triple Point Estimation.....	33
<b>4..Results .....</b>	<b>34</b>
4.1 Global versus Local Velocity Estimation .....	34
4.2 Experiment No. 2516.....	35
4.3 Experiment No. 2533.....	36
4.4 Experiment No. 2543.....	37
4.5 Experiment No. 2573.....	38
4.6 Experiment No. 2592.....	39
4.7 Result Comparison with CPSE.....	40
<b>5..Discussions .....</b>	<b>42</b>
5.1 Normalization of Data .....	42
5.2 Outliers .....	42
5.3 Effectiveness and Accuracy of Framework .....	42
5.4 Error Margins .....	42
5.5 Template Angle Acceptance.....	43
<b>6..Conclusions .....</b>	<b>45</b>
6.1 Future Work.....	45

<b>6.1.1 Code Optimization .....</b>	<b>45</b>
<b>6.1.2 Outlier Removal.....</b>	<b>45</b>
<b>6.1.3 Normalization of Data .....</b>	<b>45</b>
<b>6.1.4 Template Value Update.....</b>	<b>46</b>
<b>6.1.5 Oblique Shock Template Rework .....</b>	<b>46</b>
<b>References .....</b>	<b>48</b>
<b>Appendices .....</b>	<b>49</b>

# 1 Introduction

This chapter contains practical and background information regarding the Master's thesis.

## 1.1 Background

The thesis covers the development of a framework for automatic tracking of wave fronts in high-speed video films. One major challenge is that high-speed video films are often blurred and corrupted with noise. A series of frames are shown in Figure 1.1 to illustrate the shockwave propagation captured by the high-speed camera.

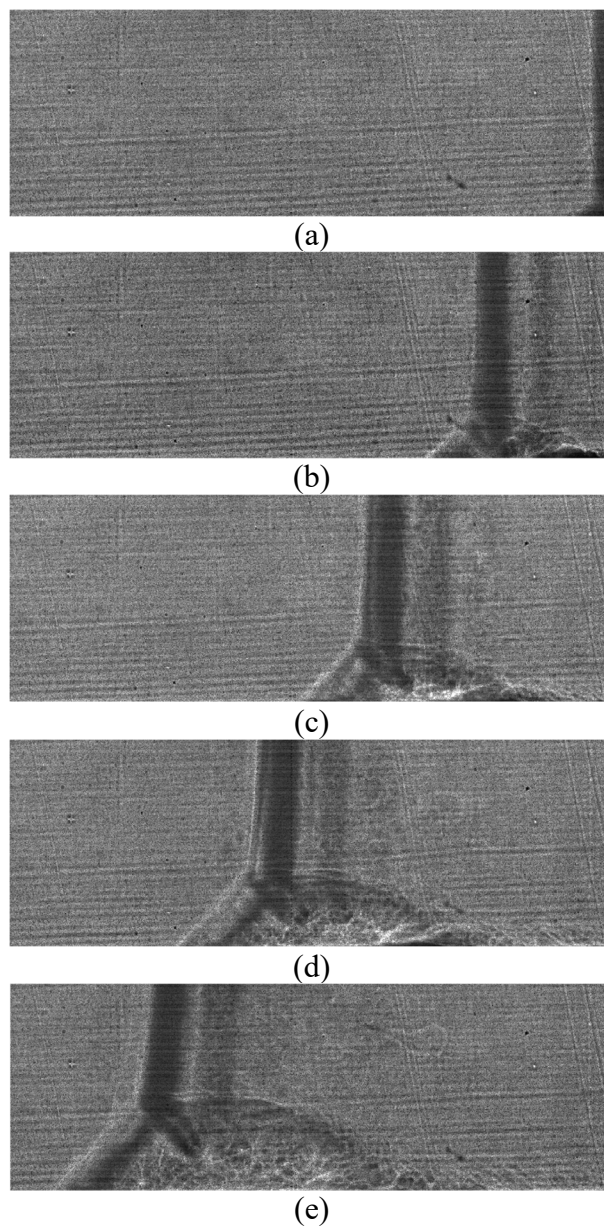


Figure 1.1: Illustration of wave front propagation through the test section of the test rig. Images (a) through (e) shows frames from 62 through 142 with a 20-frame interval between each frame.

Figure 1.2 displays a typical frame taken from one of the high-speed videos provided. Originally the frame is darker, but has been enlightened for illustration purposes. Figure 1.2 also displays elements of interest regarding the wave front.

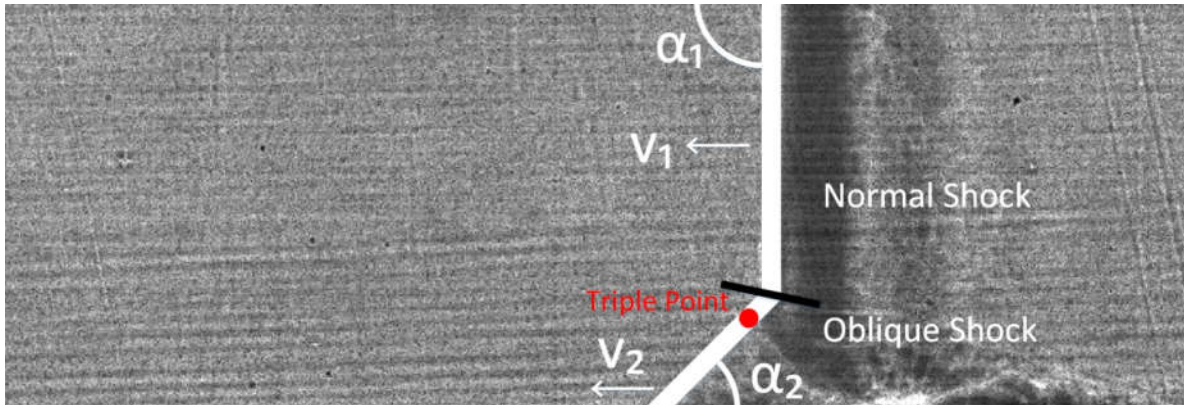


Figure 1.2: Image of a wave front traveling from the right in the picture, towards the left. Elements of interest are displayed as  $v_1$ ,  $v_2$ ,  $\alpha_1$  and  $\alpha_2$  representing the velocity and angle of the normal and OS respectively. The red dot represents the TP.

The high-speed videos usually consist of approximately 100 images, which are represented by matrices with 241 rows and 701 columns. Each matrix element represents a pixel with grayscale value between 0 and 255, representing black and white respectively.

### 1.1.1 Test Rig

The wave front is generated by a test rig where a flammable gas mixture is pumped into the shock tube, then ignited. The rest of the tube is filled with a test gas to test wave front velocities through different gases. Among the gases in which the wave front velocity is tested are Ar, N<sub>2</sub> and CO<sub>2</sub>. Figure 1.3 shows a drawing of the test rig used in production of the high-speed videos. The test section of the test rig has a window in which the high-speed camera can film the wave front at 500 000 frames per second (*fps*).

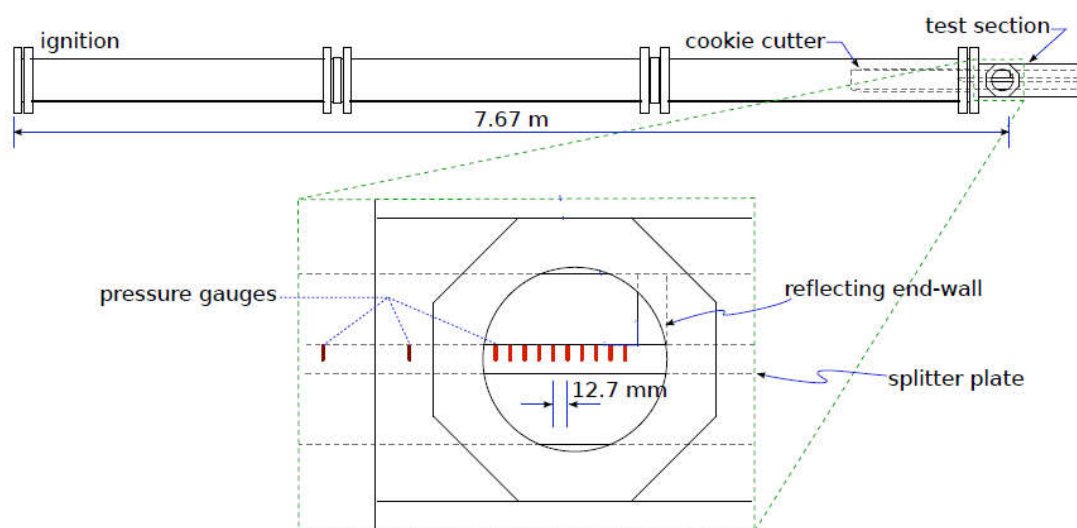


Figure 1.3: Drawing of the test rig setup for wave front creation. The splitter plate is equipped with pressure sensors which are used to calculate velocity of the OS.



## 1 Introduction

During the test procedure, the wave front hits the end-wall in the test section and reflects causing the wanted shock phenomena to occur. The splitter plate is equipped with pressure sensors which can be used to calculate the OS velocity. A closer look at the splitter plate is seen in Figure 1.4.

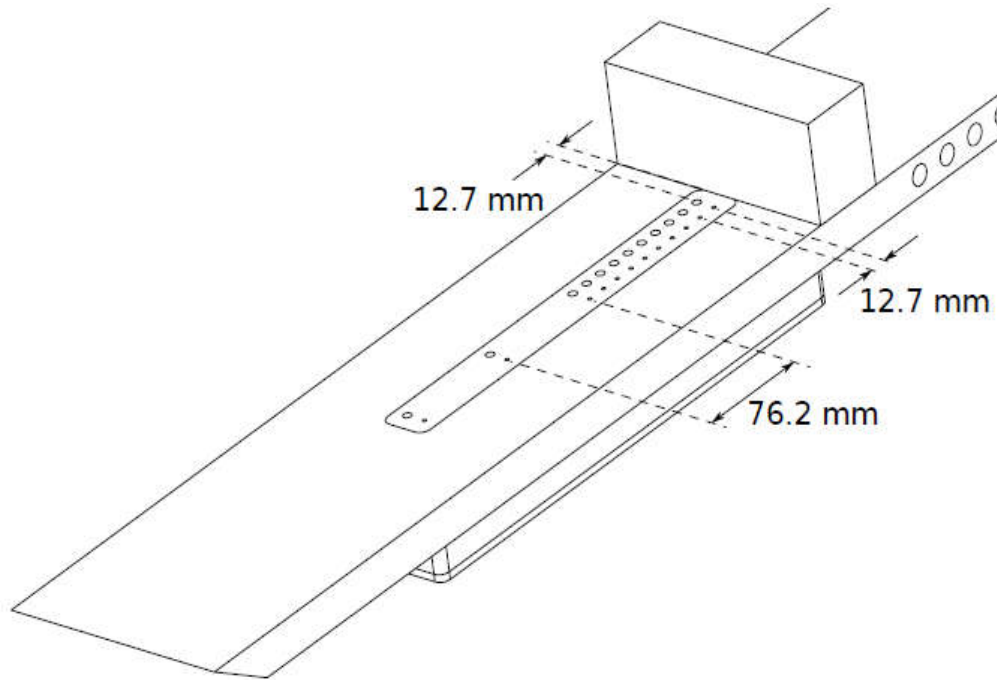


Figure 1.4: Drawing of the splitter plate which resides within the test section of the test rig.

The splitter plate contains pressure sensors which are placed in the holes in the splitter plate. This meaning the accuracy of which the holes have been drilled has an impact when calculating wave front velocities, as the wave front velocities are sometimes upwards of  $400 \left[ \frac{m}{s} \right]$ , depending on the test gas. Having velocities calculated through image processing in addition to pressure sensors creates a way to compare results.

## 1.2 Task Description

This report covers the development of a framework for wave front tracking of explosions. The high-speed videos provided are a joint work between Combustion, Process Safety and Explosions research group at USN and Caltech.

The developer will work with the Combustion, Process Safety and Explosions research group and adapt the goal of the framework to satisfy the information requirement by the research group in collaboration with the supervisor.

The goals of the framework to be developed are:

- Semi-automatic wave front tracking
- Triple point
  - Coordinate estimation

- Velocity estimation
- Normal and oblique shock
  - Angle estimation
  - Velocity estimation

### **1.3 Constraints and Extensions**

Code optimization is not prioritized in this task as framework performance and success is not measured in speed, but rather the accuracy of which the wave fronts are tracked.

Outliers are expected to occur for the wave front tracking. The action of removing outliers is its own field of expertise all together and would make for a Master's thesis on its own. Therefore, outlier removal is considerably simplified as it is not within the scope of this thesis.

The task has been extended to also include tracking of TPs in the high-speed wave front videos.

## 2 Literature Study

The specifics of this thesis are very narrow, making relevant literature a challenge to uncover. Therefore, this literature study is aimed more towards finding information regarding edge detection in images.

### 2.1 Image Thresholding

The threshold method, Otsu, was studied [1]. The method is included in MATLAB [2]. The Otsu threshold technique splits the image into black and white using a threshold level. This is done attempting to separate noise from the elements of interest, which in this case is a shock wave. Figure 2.1 shows a raw image of a wave front.



Figure 2.1: Raw wave front image of which the Otsu threshold technique is applied.

Using a histogram, as shown in Figure 2.2, the Otsu technique has set a threshold level at pixel value 28.25 for the given image.

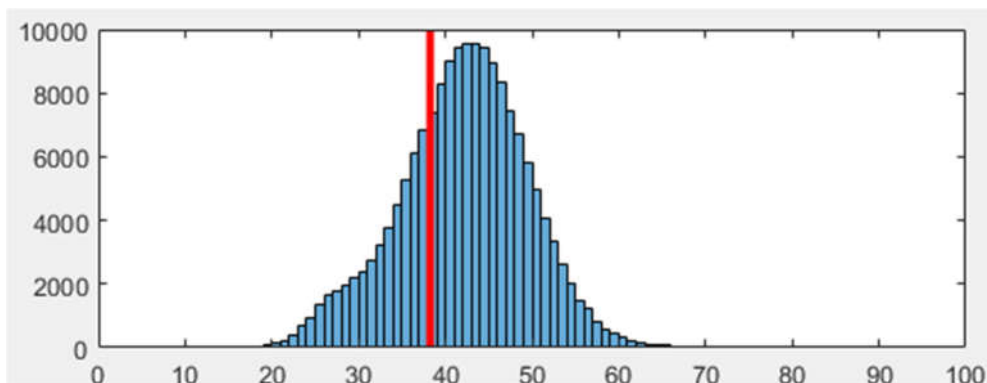


Figure 2.2: Histogram of the raw wave front image, Figure 2.1, where the x- and y-axis represents pixel value (brightness) and total number of pixels at that value, respectively. The redline represents the Otsu threshold level at 28.25.

By maximizing  $\eta$ , a threshold level  $k$  is found, see Formula (2.1).

$$\max(\eta(k)), \text{ where } 1 \leq k < L$$

and,

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_T^2} \quad (2.1)$$

Where

$k$  = threshold level

$L$  = total number of threshold levels

$\sigma_B(k)$  = the variance of pixel values below threshold  $k$

$\sigma_T$  = total variance of pixel values in the image

Figure 2.3 shows the output image of the Otsu threshold technique applied on the raw wave front image.

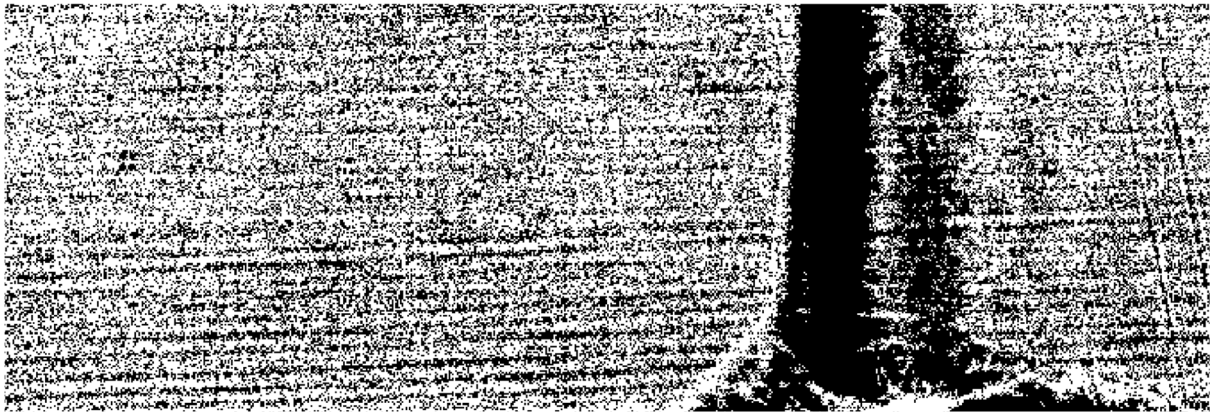


Figure 2.3: Output image of the Otsu threshold technique.

## 2.2 Edge Detection

As the wave fronts appears as vague edges in the high-speed video films, edge detection may seem feasible. MATLAB which is the preferred computational tool for this thesis already contains several edge detection techniques.

Several existing edge detection techniques were tested [3]:

- Canny
- Sobel
- Roberts
- Prewitt
- LoG (Laplacian of Gaussian)

The results of the edge detection tests are seen in Figure 2.4.

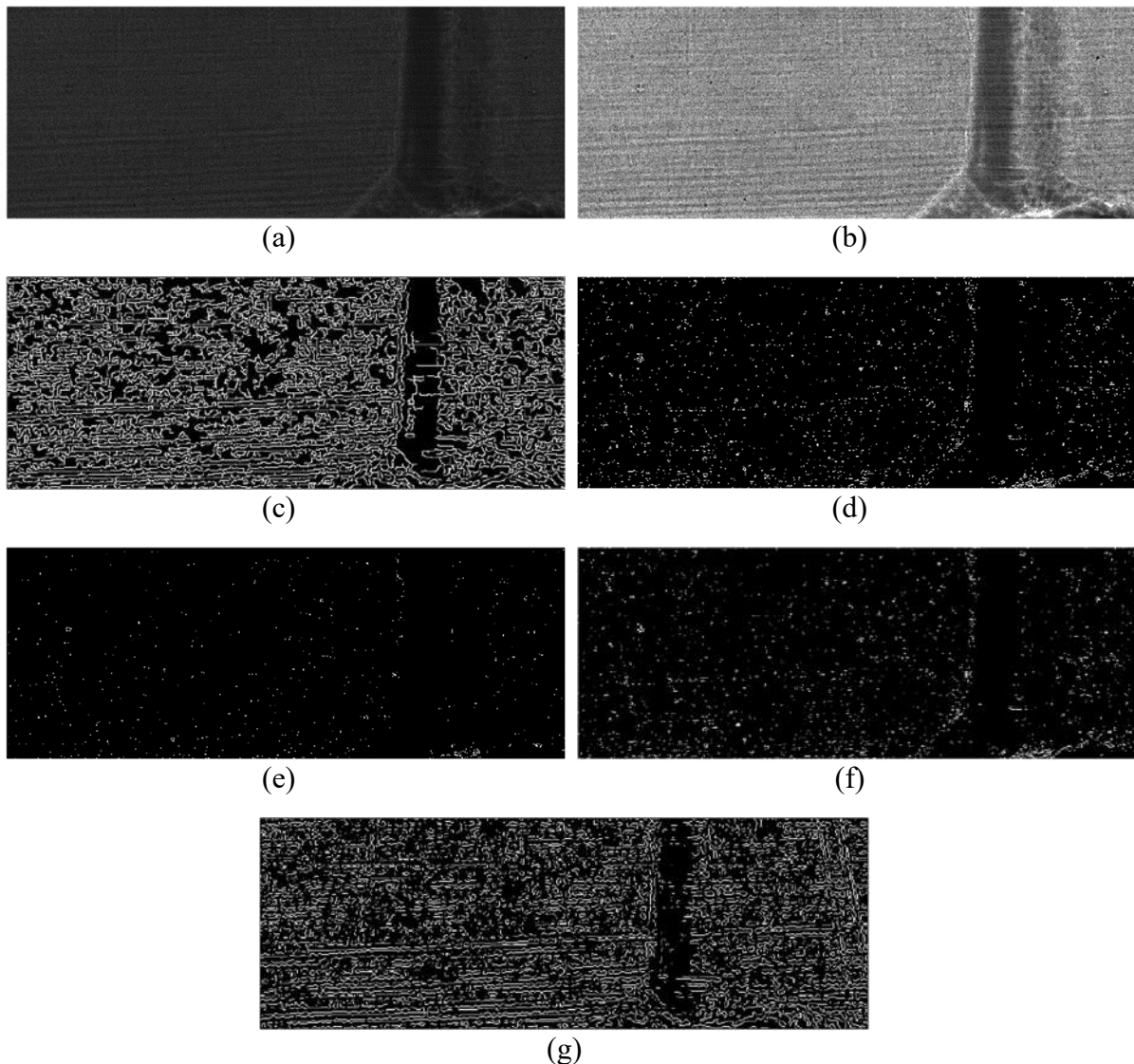


Figure 2.4: Comparison of various edge detection techniques: (a) raw image, (b) enlightened image, (c) Canny method, (d) Sobel method, (e) Roberts method, (f) Prewitt method and (g) LoG method.

From these results, it is concluded that a technique which is less sensitive to background interference is needed.

An issue regarding these edge detectors is that too many edges are found. This includes turbulence as well as the back end of the wave front. Figure 2.4 shows a comparison of the mentioned techniques applied to the same wave front image.

Importantly, within each frame the background and wave front merges, meaning they have the same pixel values at certain points, as illustrated in Figure 2.5. Moreover, the pixel values of the image background vary from one frame to the next, making tracking of the wave front even more intricate. Other videos have slightly different pixel values, however, the effect of background interference is similar. Figure 2.5 displays how the background and the wave front in some cases blends together, which makes edge detection difficult.

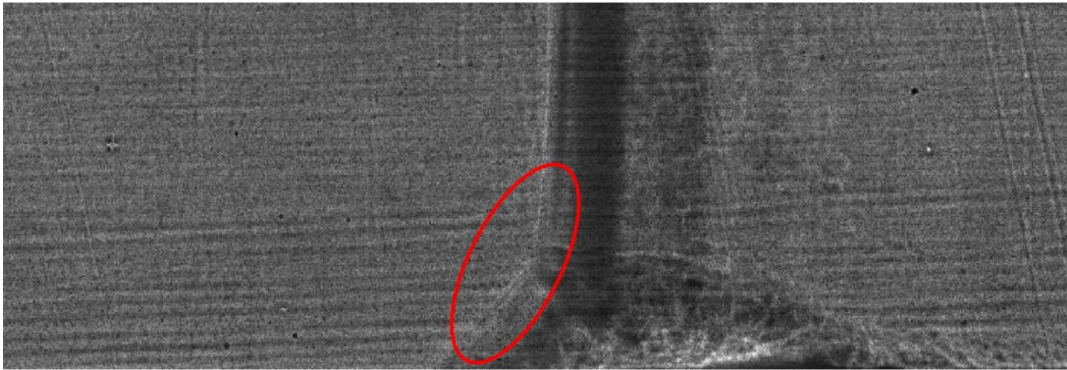


Figure 2.5: Illustration of how the background “merges” with the wave front and becomes almost inseparable from the background, especially in the transition between the normal and oblique shock.

A technique which compares a larger area to find vague edges is investigated.

## 2.3 Template Matching

Template matching can be used as an edge detector [4][5], and a wave front may be classified as an edge. In template matching, a template is run across an image, row by row, while the difference between the template and the image is calculated for each iteration. In Figure 2.6 template matching for wave front tracking is illustrated.

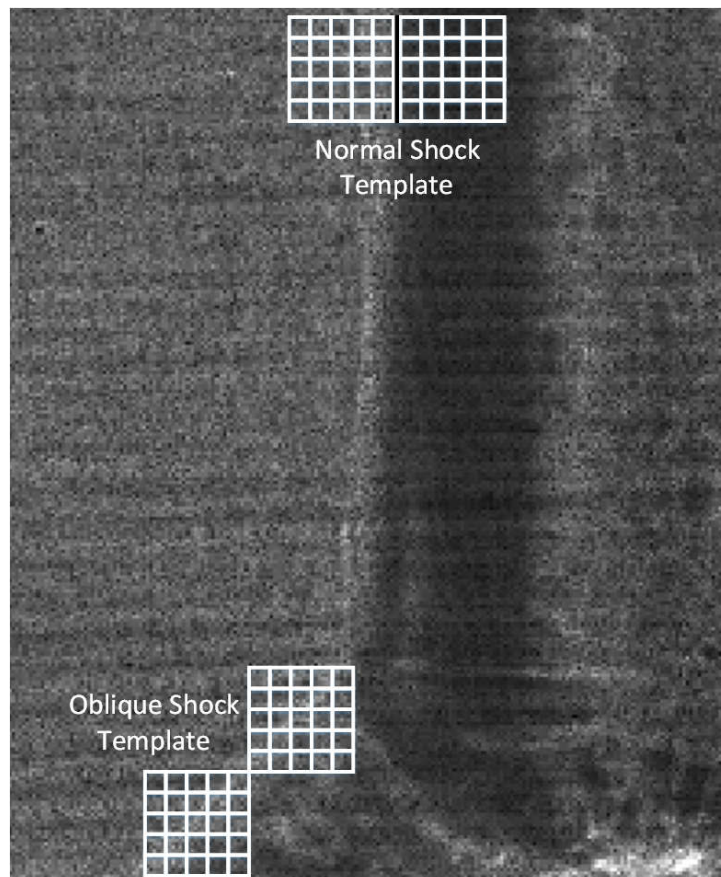


Figure 2.6: Template setup for wave front tracking. Templates are not to scale.

## 2 Literature Study

The templates have a pre-allocated value where for the NS template, the left template average should match that of the background and the wave front, while the right template should match that of the darker section of the wave front. The OS template should match the brighter “foot” of the wave front to obtain an accurate fit match.

The result of the NS template is seen in Figure 2.7, and the result for the OS template is seen in Figure 2.8.

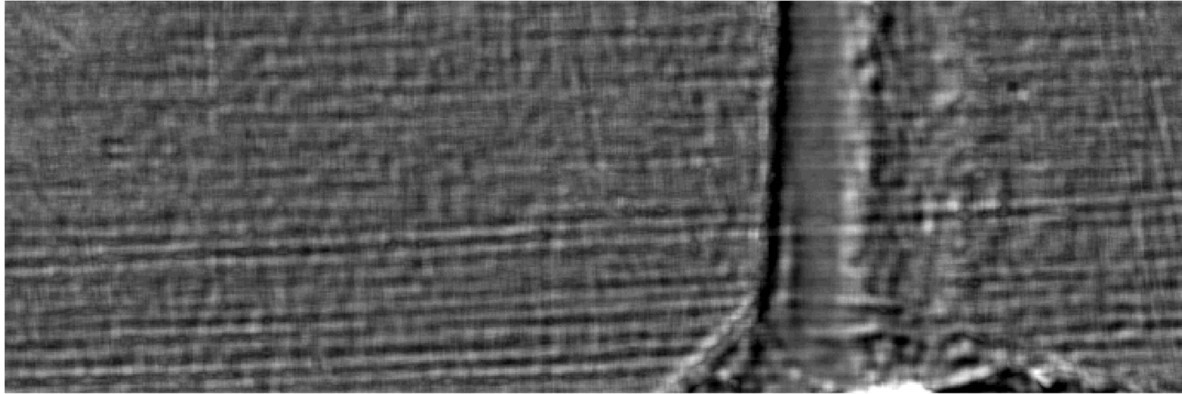


Figure 2.7: Error matrix generated by the NS template.

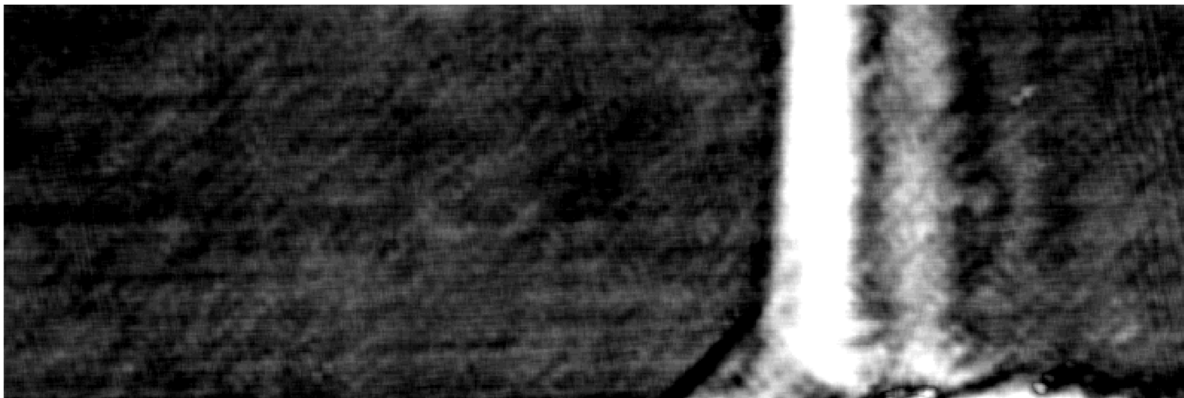


Figure 2.8: Error matrix generated by the OS template.

The result of the template matching is two error matrices where the smallest error represents the darkest pixels. One advantage with this template matching technique is that the best template matching result for every row is represented as the darkest pixel in every row, making obtaining the coordinates of the best template match straightforward.

The ability to obtain a good match on the vague wave fronts is achieved by using templates which span a wide area. In this case, a template consists of two 5-by-5 pixel matrices placed side by side, as shown in Figure 2.6.

## 2.4 Watershed

The Watershed technique treats the image as a landscape where bright pixels represents tall elevations, and dark pixels represents low elevations. This creates “ridge lines” and “basins”, the landscape is then filled with water having dark pixels filled first as these are at low elevation [6].

## 2 Literature Study

A drawback of watershed is that pre-processing of the image is often needed to avoid oversegmentation. Which can be seen in Figure 2.14. The process of applying the watershed technique onto a raw wave front image Figure 2.9, can be seen through Figure 2.10, Figure 2.11 and Figure 2.12.



Figure 2.9: Raw wave front image of which the watershed technique is to be applied.

The raw image if first background subtracted and filtered.

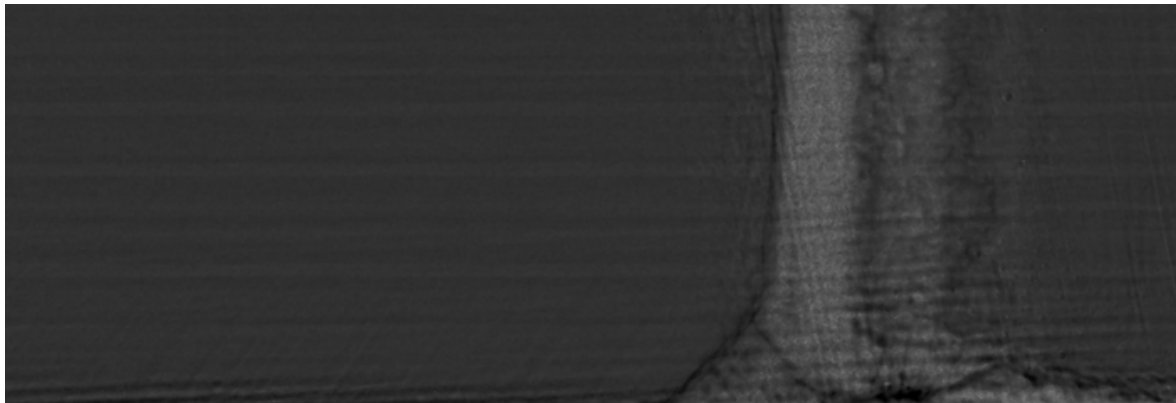


Figure 2.10: Image after background subtraction and filtering.

Once background subtraction and filtering has been applied, the image is thresholded, as shown in Figure 2.11.



Figure 2.11: Image after a threshold technique is applied.

After the threshold technique has been applied to the wave front image, watershed is applied.





Figure 2.12: Thresholded wave front image after Watershed is applied.

Figure 2.12 shows the result of a watershed implementation.

If the noise is not suppressed from the wave front image before thresholding, see Figure 2.13, over-segmentation appears as shown in Figure 2.14.

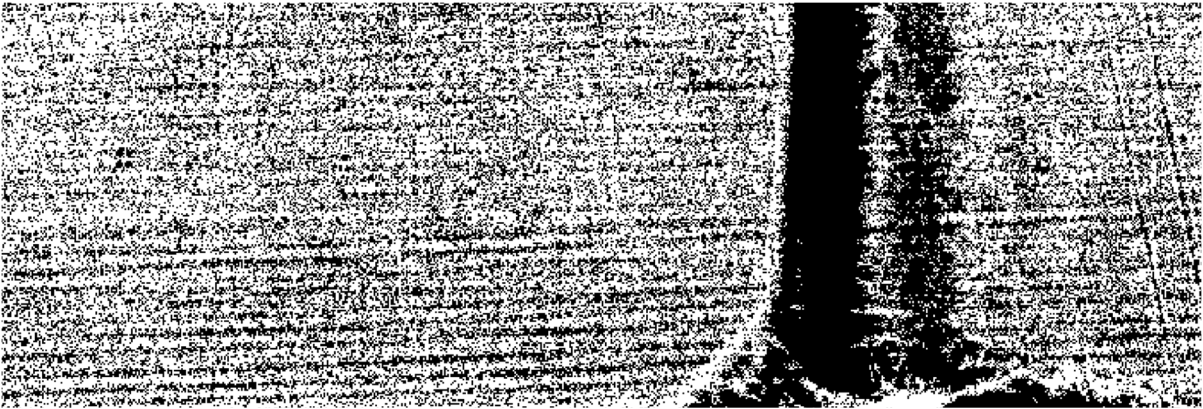


Figure 2.13: Output image of the Otsu threshold technique without any filtering.

It is seen that the thresholded image now contains much more noise, as seen in Figure 2.13, compared to that of Figure 2.11. The result is an over-segmented watershed result, as seen in Figure 2.14.

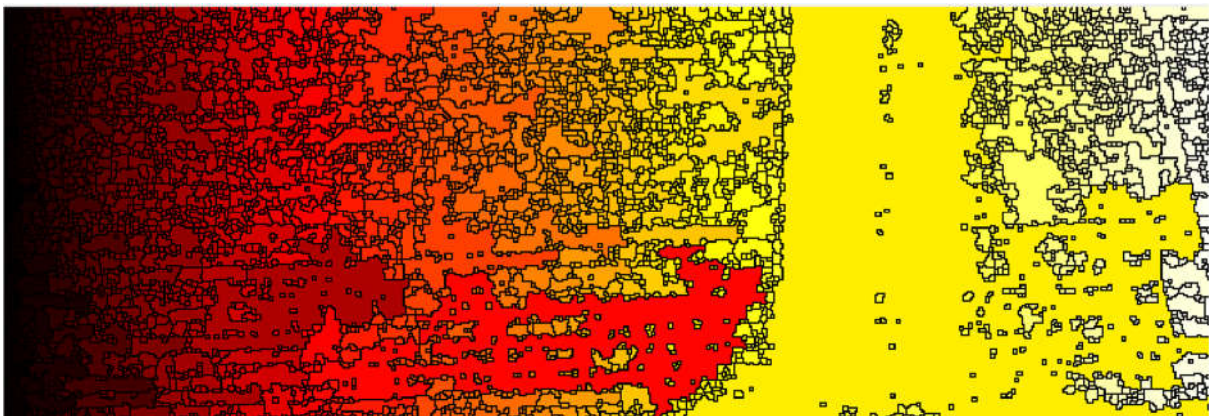


Figure 2.14: Watershed applied on thresholded image with no noise filtering.

## 2.5 Fourier Transform in Image Processing

The Fourier Transform for image processing can be used to find the frequencies in which the interference resides and then suppress these frequencies [7]. Then, once the interference is suppressed, the inverse transform is applied and the image should appear with reduced interference. Figure 2.15 shows a raw image of a wave front of which the Fourier transform is to be applied. The output image is shown in Figure 2.16.



Figure 2.15: Raw image of which the Fourier transform is to be applied.

Suppressing interference using the Fourier transform is dependent on having the interference occur at specific frequencies. Otherwise, isolating and removing the interference proves difficult.

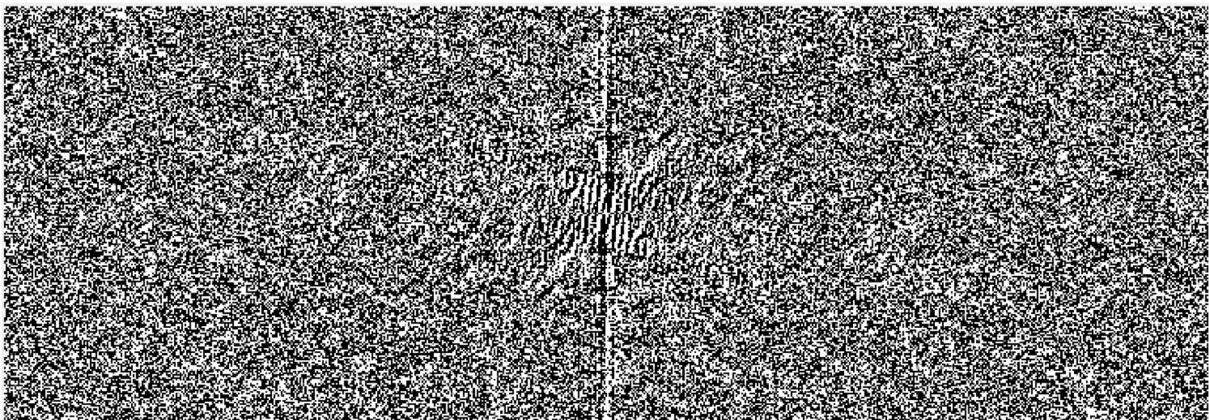


Figure 2.16: Output image from the Fourier transform.

It can be seen in Figure 2.16 that distinguishing the shockwave from the background noise may be difficult at a frequency plane in this case. This result indicates that there are no obvious frequencies of which the wave front appears separated from the background interference.

## 3 Methods

This chapter explains the methods investigated for achieving satisfactory wave front and TP tracking, in addition to angle and velocity estimation. The method this thesis will be focusing on is template matching.

### 3.1 Outlier Frames

It was found that every 10<sup>th</sup> image (i.e. no.: 1, 11, 21, 31 etc.) in the high-speed video films were severely corrupted by noise and blur compared to the rest of the film, these were removed and are not expected to have significant influence of the final results. The removed frames must, however, be considered when estimating velocities, as the number of frames dictates the elapsed time of a wave front over a given distance.

### 3.2 Wave Front Template Matching

Due to prior knowledge about the wave front obtained through studying the high-speed videos, the NS and OS appears to occur at approximately 90 and 45-degree angles respectively. Template matching described in Chapter 2.3 is evaluated as a good technique for finding such elements in an image. For the present high-speed videos, the edge detection techniques included in MATLAB can not be used straightforward, as illustrated in Chapter 2.3, Therefore, a new method based on template matching is developed.

As mentioned, the wave fronts appear to have a near 90-degree NS and a near 45-degree OS. To effectively track the two shocks, two templates are implemented, each specialized in finding either the NS or OS.

The high-speed videos acquired seem to contain Gaussian noise which the templates are required to handle.

#### 3.2.1 Normal Shock Template

The template described in this chapter tracks the vertical portion of the wave front.

Two 5-by-5 pixel matrices are implemented side by side and steps through the image row by row, comparing their pre-allocated template values with the average pixel value within each matrix, as shown in Formula (3.1).

$$E_{k,l} = |\bar{L}_{k,l} \quad L_{pre}| + |\bar{R}_{k,l} \quad R_{pre}|$$

$$\text{for } 7 \leq k \leq m \quad 7 \text{ and } 7 \leq l \leq n \quad 7$$

$$\bar{L}_{k,l} = \frac{\sum_{i=k-6}^k (\sum_{j=l-2}^{l+2} (a_{i,j}))}{25}$$

$$\bar{R}_{k,l} = \frac{\sum_{i=k+6}^k (\sum_{j=l-2}^{l+2} (a_{i,j}))}{25}$$
(3.1)

Where

$E$  = Error matrix generated by the template matching

$a$  = the pixel value in the image

$\bar{L}_{k,l}$  = the mean of the left 5-by-5 matrix

$\bar{R}_{k,l}$  = the mean of the right 5-by-5 matrix

$m$  = number of rows in the image

$n$  = number of columns in the image

$L_{pre}$  = pre-allocated average value for left 5-by-5 matrix

$R_{pre}$  = pre-allocated average value for right 5-by-5 matrix

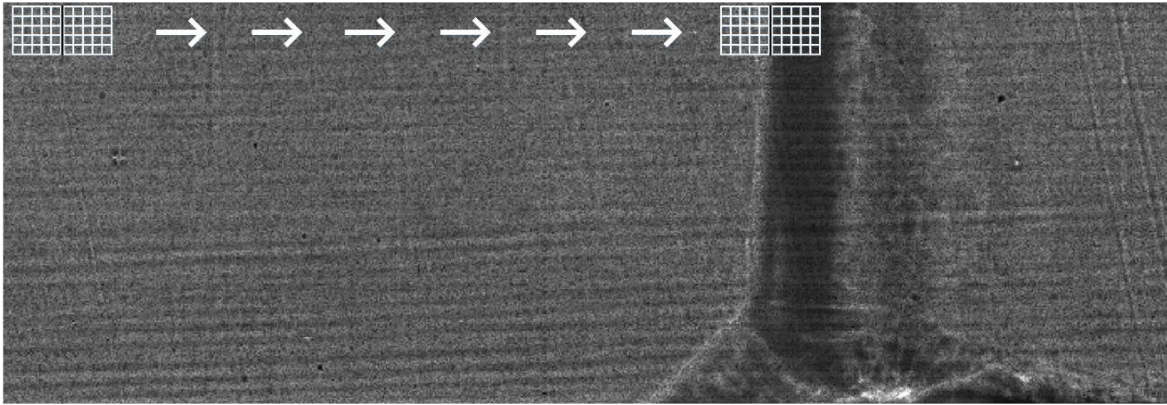


Figure 3.1: Illustration of template stepping from left to right averaging all the pixels within each 5-by-5 matrix (matrices not to scale).

All the results which are obtained through the template matching is stored in the error matrix,  $E$ , and can be visualized in a separate image, see Figure 3.2. Pixel values ranges from 0 to 255 which represents black and white respectively.

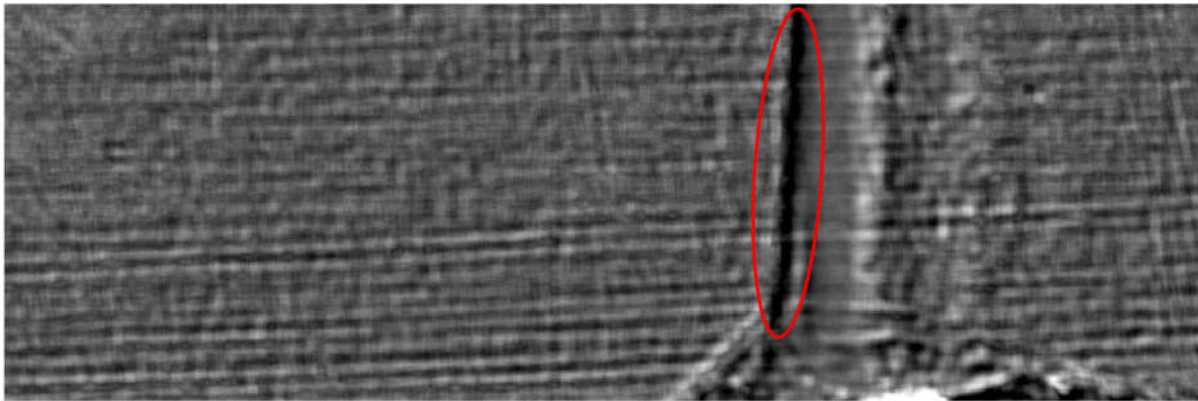


Figure 3.2: Visualization of deviation between the NS template and the raw image. A dark ridge is prominent where the wave front is expected to be, indicated by the red ellipse. Smaller deviations are represented as darker pixels, which represents a good match.

When the NS template has run through the image, finding the wave front becomes a minimization problem. This can be solved by finding the pixel with the lowest value in  $E$  on each row, as illustrated in Figure 3.3.

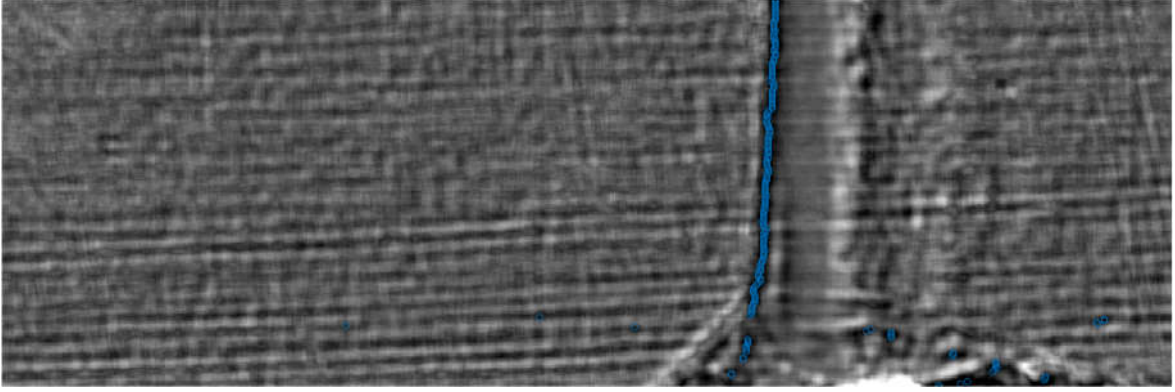


Figure 3.3: The minimization problem illustrated with blue circles indicating the lowest pixel value on each row.

Figure 3.3 illustrates how the vertical template tracks the wave front, it also illustrates how the NS template is inadequate in tracking the OS. A second template is introduced to achieve adequate OS tracking.

### 3.2.2 Oblique Shock Template

The template described in this chapter tracks the oblique portion of the wave front.

The OS template is similar to the NS template, but skewed in order to track a 45-degree feature, as illustrated in Figure 3.4 and Formula (3.2).

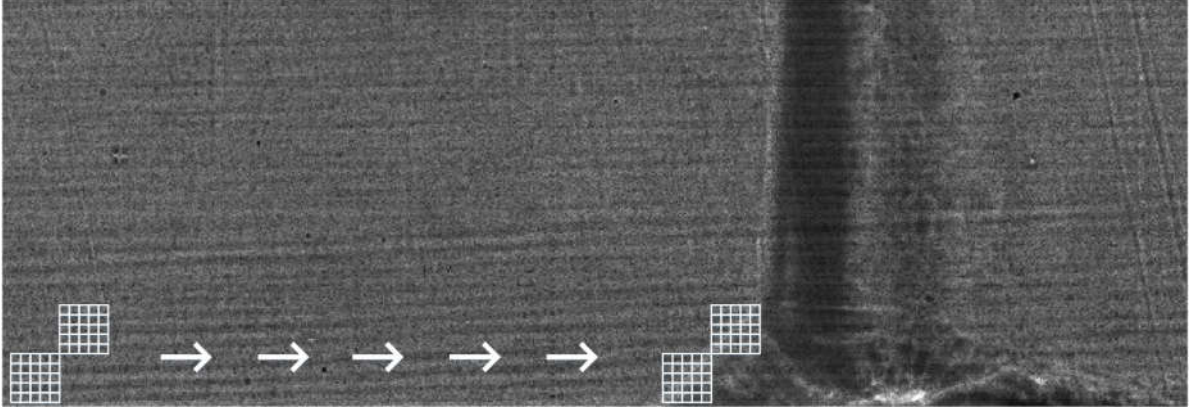


Figure 3.4: Illustration of OS template on a raw image.

$$E_{k,l} = |\bar{L}_{k,l} \quad L_{pre}| + |\bar{R}_{k,l} \quad R_{pre}|$$

for  $7 \leq k \leq m - 7$  and  $7 \leq l \leq n - 7$

$$\bar{L}_{k,l} = \frac{\sum_{i=k-5}^k (\sum_{j=l-5}^l (a_{i,j}))}{25} \quad (3.2)$$

$$\bar{R}_{k,l} = \frac{\sum_{i=k}^{k+5} (\sum_{j=l}^{l+5} (a_{i,j}))}{25}$$

Where

$E$  = Error matrix generated by the template matching

$a$  = is the pixel value in the image

$\bar{L}_{k,l}$  = the mean of the left 5-by-5 matrix

$\bar{R}_{k,l}$  = the mean of the right 5-by-5 matrix

$m$  = number of rows in the image

$n$  = number of columns in the image

$L_{pre}$  = pre-allocated average value for left 5-by-5 matrix

$R_{pre}$  = pre-allocated average value for right 5-by-5 matrix

All the results which are obtained through the template matching is stored in the error matrix,  $E$ , and can be visualized in a separate image, see Figure 3.5.

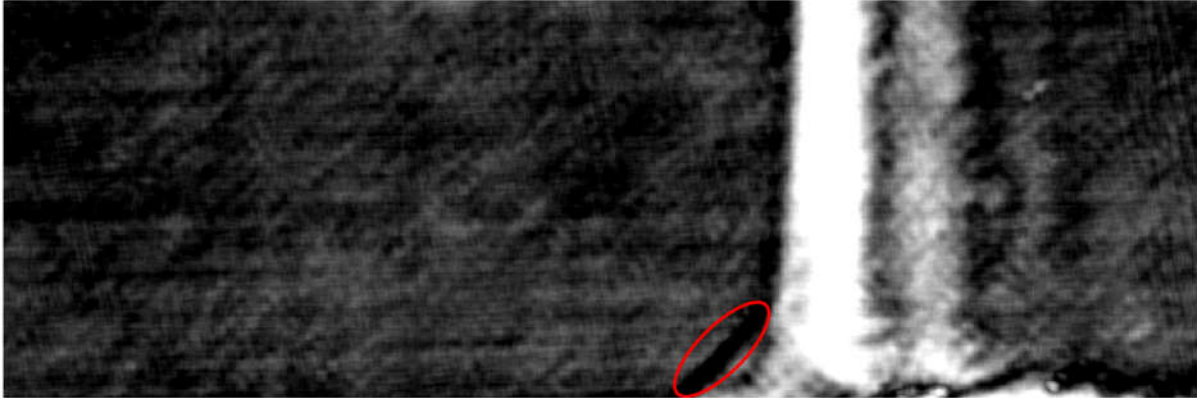


Figure 3.5: Visualization of deviation between the OS template and the raw image. A dark ridge is prominent where the wave front is expected to be, indicated by the red ellipse. Smaller deviations are represented as darker pixels, which represents a good match.

When the OS template has run through the image, finding the wave front becomes a minimization problem. This can be solved by finding the pixel with the lowest value in  $E$  on each row, as illustrated in Figure 3.6.

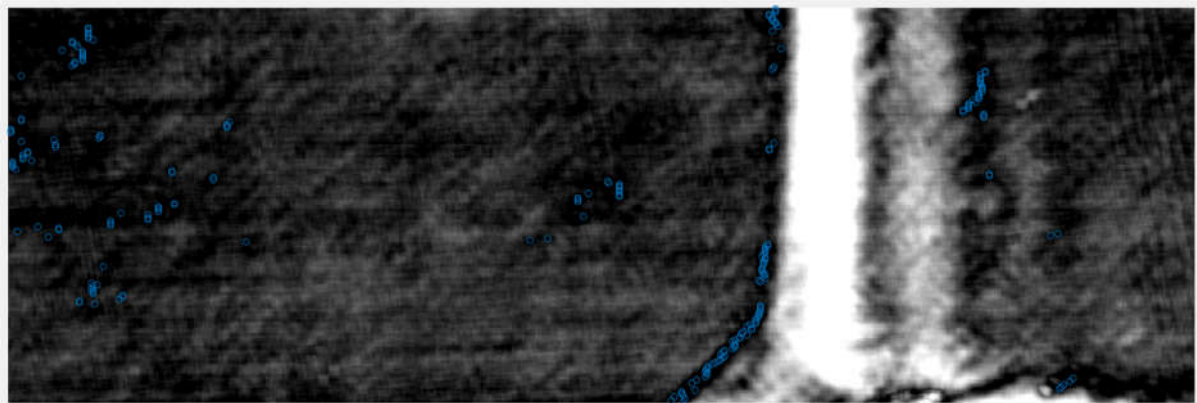


Figure 3.6: The minimization problem illustrated with blue circles indicating the lowest pixel value on each row.

Figure 3.6 illustrates how the OS template tracks the OS. It is also illustrated how the OS template poorly tracks the NS, underlining how a combination of the two templates is necessary.

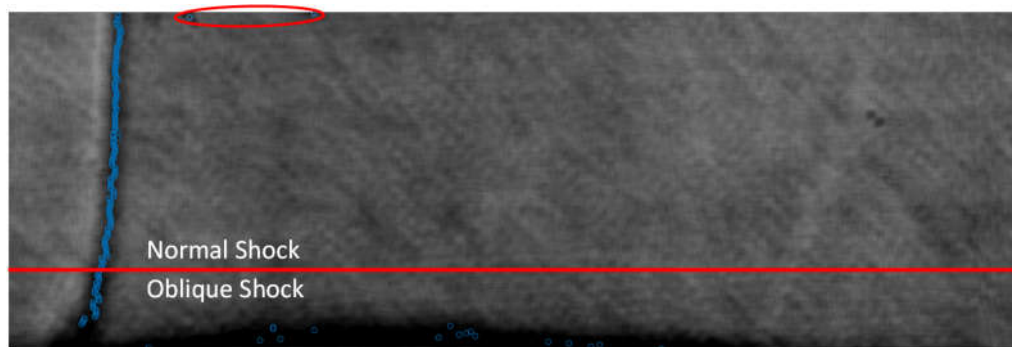
After the OS template has run through the image, the basis is set for further operations. The OS tracking is further improved as described in Chapter 3.6, where the OS template is run a second time on a downsized image.

### 3.2.3 Outlier Correction

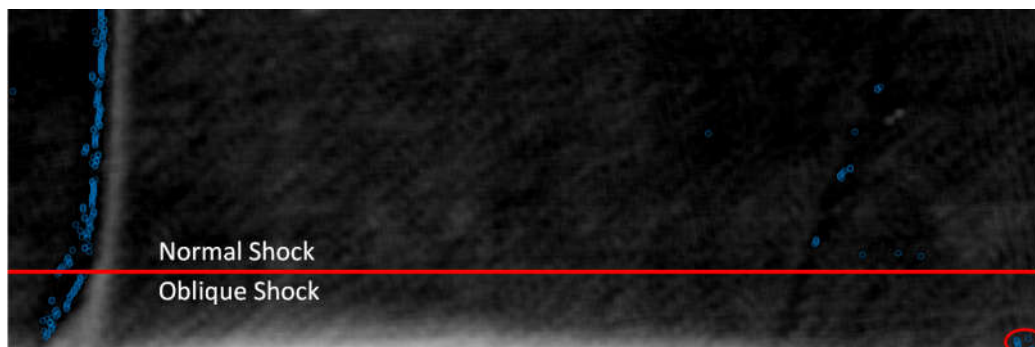
As seen in Figure 3.3 and Figure 3.6 outliers occur, i.e. some blue circles are mislocated far from the wave front. Such outliers must be handled to minimize further disturbance in the wave front tracking. A simple function is implemented which removes outliers which are located a certain threshold away from the bulk of the points found by the normal and OS template. This ensures removal of the most extreme outliers which interfere the most with the wave front tracking.

## 3.3 Sliding Window

The wave front tracking has some challenges regarding background interference and turbulence which occurs in the front and in the wake of the wave front, resulting in outliers being generated, as seen in Figure 3.7.



(a)



(b)

Figure 3.7: (a) and (b) shows normal and oblique shock template generating outliers. The outliers are marked by the ellipses.

The NS template is not expected to find the entire front, including the oblique portion of the wave front, meaning the NS template is expected to have outliers in the OS portion of the wave front.

A method was implemented to reduce the number of interfering outliers. The sliding window was implemented such that the user selects the points at which the wave front starts in the first frame, and the width of the wave front in the last frame.

An illustration of the method is displayed in Figure 3.8.

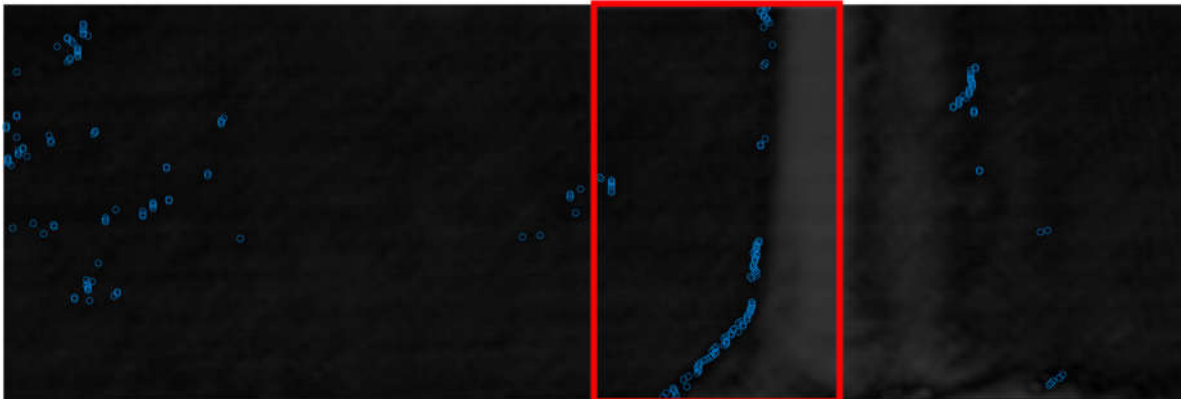


Figure 3.8: The sliding window forces the vertical and OS template matching to search within the sliding window.

The result is a sliding window which follows the wave front from one frame to the other, reducing the number of outliers as the templates matching no longer searches every image in its entirety. The result of the OS template run within the sliding window is seen in Figure 3.9.

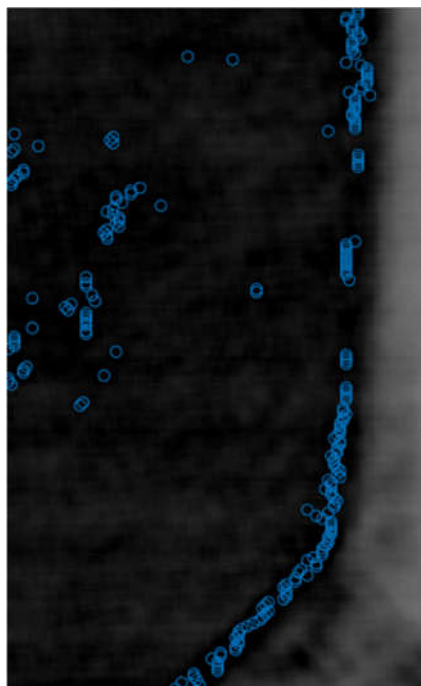


Figure 3.9: Result of template matching within the sliding window.

### 3.4 Segmented Regression: Normal and Oblique Shock

This chapter describes how Segmented Regression is used to find the optimal fit for both NS and OS. An extended Segmented Regression was developed to allow for applying Segmented Regression to two separate datasets in parallel.



### 3.4.1 Segmented Regression

Segmented Regression is a method used to fit multiple lines or curves to a dataset by splitting the dataset into segments and apply least squares fit to the segmented datasets[8]. The data is segmented to achieve a better fit. This technique works best if the data points appear clustered into different groups [9]. A simple Segmented Regression technique uses two line segments calculating the combination of least square fit which presents the least error. An illustration of a Segmented Regression result is shown in Figure 3.10.

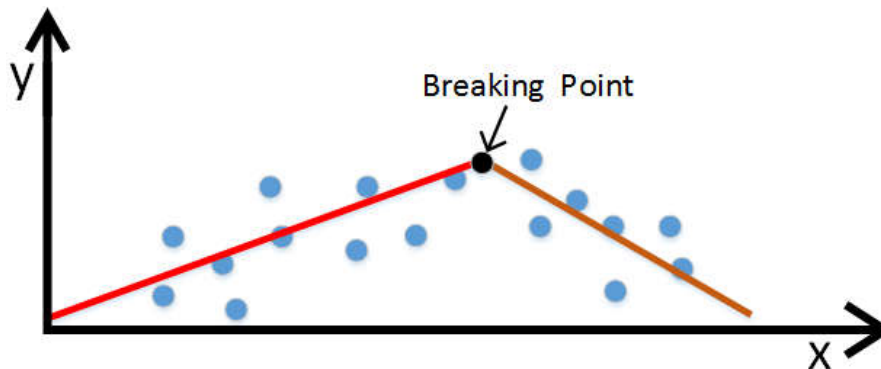


Figure 3.10: Illustration of a typical Segmented Regression result. The red and orange line represents the least square fit for each segment of the dataset.

The two lines generated by the Segmented Regression method are joined at the breaking point, which is the optimal point of dataset segmentation.

The problem presented in this thesis requires the Segmented Regression method to find the optimal point of segmentation of two separate datasets generated by the NS and OS templates. The Segmented Regression method needs to be extended to handle two separate datasets in parallel.

### 3.4.2 Extended Segmented Regression

The NS and OS estimation, as well as the breaking point between the NS and OS are calculated using Segmented Regression. However, for the problem presented in this thesis, the Segmented Regression uses two line segments which in combination with two different datasets yields the least combined error. The point of which the combined error is the least between the two datasets is called the breaking point, and is what separates the NS from the OS. As this task requires the Segmented Regression to work with two datasets in parallel, an extended Segmented Regression method is developed. The result of the segmented regression is shown in Figure 3.11.

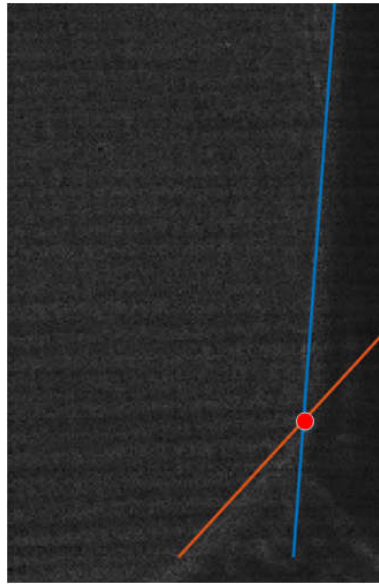


Figure 3.11: Result of segmented regression using two separate datasets and finding the combined least error. The red dot represents the breaking point.

The two datasets used to obtain the result shown in Figure 3.11 are shown in Figure 3.12.

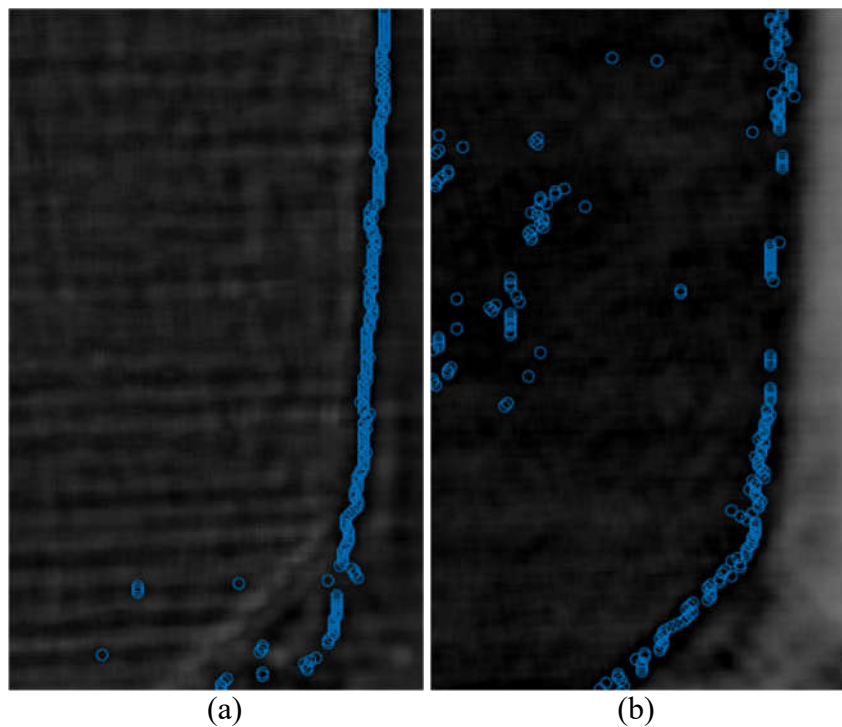


Figure 3.12: The two datasets which segmented regression is applied to. (a) is the result of the NS template matching, while (b) is the result of the OS template matching. Both template matches are applied within the sliding window, described in Chapter 3.3.

The Segmented Regression technique used is a brute force approach where the first iteration consists of creating two line segments using only the two first points of the first dataset and the remaining points of the second dataset. It then iterates through each dataset using more points from the first dataset, and fewer points of the second dataset. The error is then calculated for

### 3 Methods

both datasets and line segments, only calculating error between the line segment and the points it has been fitted to. The combination of line segments which gives the least combined error is selected, yielding a breaking point at the point of which the line segments cross. The extended Segmented Regression brute force approach is described in Formula (3.3).

$$S_i = \sum_{j=1}^i (\hat{y}_j - (a\hat{x}_j + b))^2 + \sum_{j=i+1}^n (\bar{y}_j - (c\bar{x}_j + d))^2 \quad (3.3)$$

*for*  $2 \leq i \leq n - 2$

Where

$S_i$  = the combined error of both segments

$n$  = number of rows in the image

$\hat{y}_j, \hat{x}_j$  = the coordinates marked by the NS template

$\bar{y}_j, \bar{x}_j$  = the coordinates marked by the OS template

$a, b, c, d$  = the coefficients in the two line segments representing the NS and OS

The wave front moves in the x-direction, therefore the least square error must be calculated in the x-direction as well, as shown in Figure 3.13.

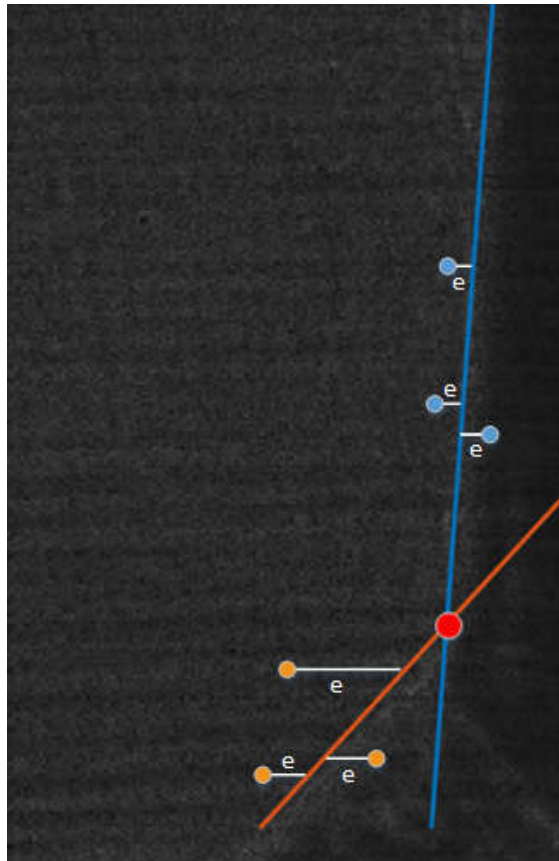


Figure 3.13: Result of segmented regression applied on normal and oblique shock template. “e” represents error between the proposed line and each point. The error is calculated in x-direction. Orange dots represent OS data points, while blue dots represent NS data points.

### 3 Methods

As shown in Figure 3.14, if the error isn't calculated in the x-direction and rather in the y-direction the increase in error would force the least square fit to generate almost horizontal wave front estimations. Making correct error calculation a critical factor to achieve a correct estimation of the wave front.

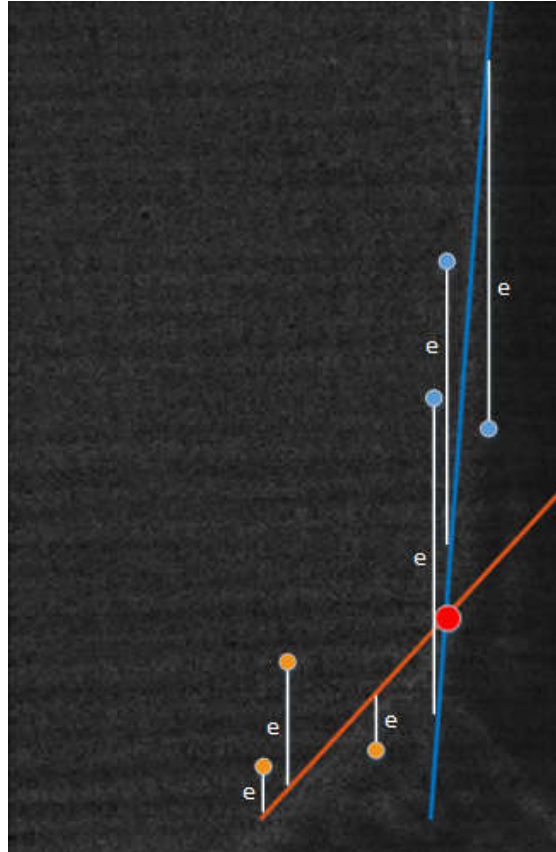


Figure 3.14: Illustration of how the error would be calculated in the y-direction. “e” represents the error to the least square fit. Orange dots represent OS data, while blue dots represent NS data.

## 3.5 Normal Shock Estimation

When the merging point is found, the NS is re-generated by using the merging point and a suitable number of points for the least square fit, 100 points in this case, from the merging point and upwards. The resulting shock is shown in Figure 3.15.

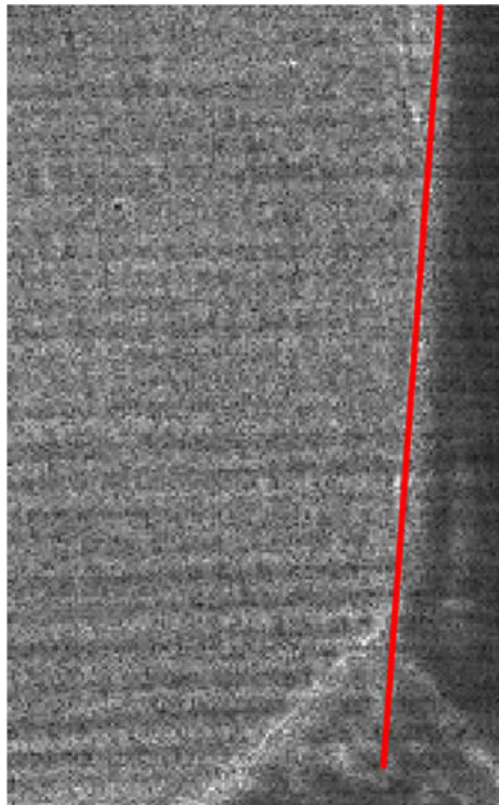


Figure 3.15: Result of re-estimation of the NS using 100 points from the merging point and upwards.

The reason to re-estimate the NS is to estimate the shock as close to the TP as possible. Meaning estimating the NS and OS from the TP and outwards is preferred as the TP is an element of interest. Estimating both shocks locally around the TP should in turn provide a good base for the estimation of the TP.

### 3.6 Oblique Shock: Secondary Tracking

Once the merging point is set, the OS template is run a second time in a cropped frame. The frame is set from the merging downwards, and leftwards. As seen in Figure 1.1, the OS always appears to the left of the NS.

An illustration and a result of the secondary OS template run is seen in Figure 3.16.

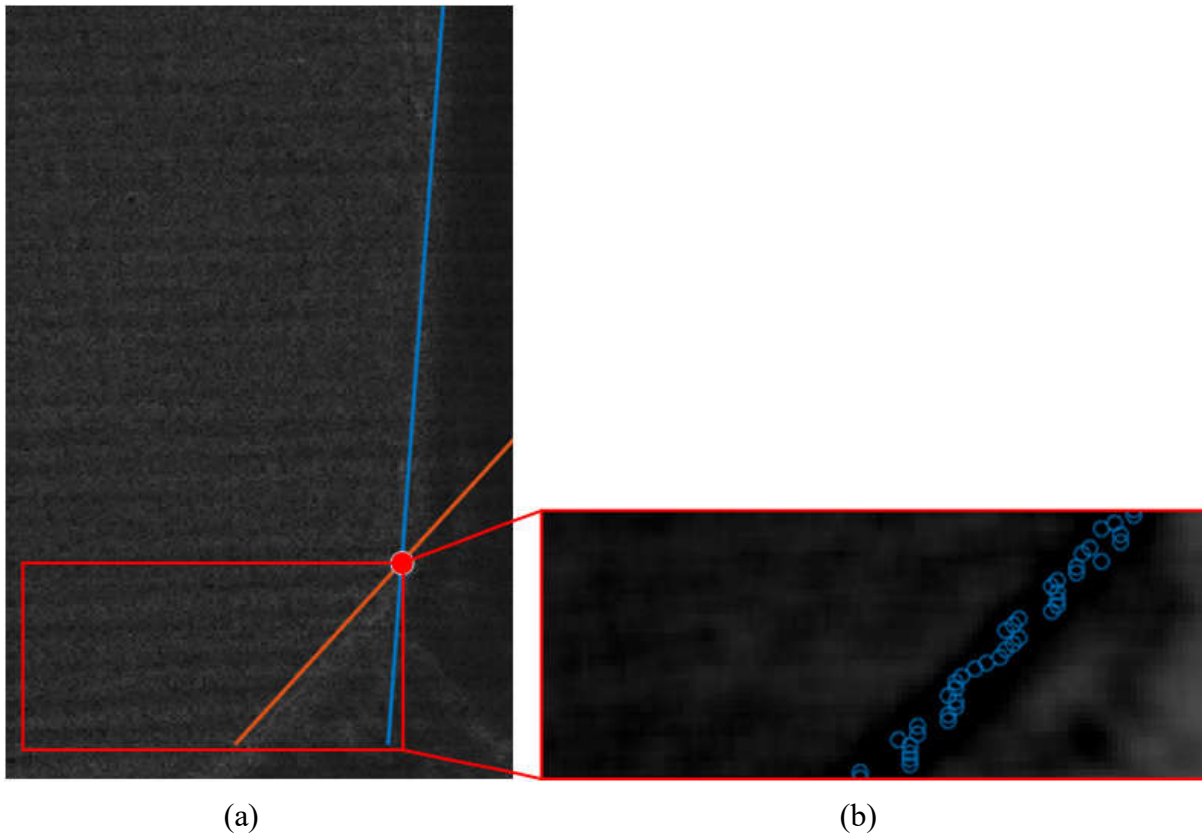


Figure 3.16: Result of OS template run a second time in a reduced frame. (a) represents the results of the image of which the OS template is run a second time. (b) represents the results of the OS template after the second run. Note that no outliers are present in the dataset shown in (b).

The search result from this operation has proven to be accurate to the extent that a least square fit applied to this dataset is used as the OS. The reason for running the OS template at such a cropped frame is the accuracy achieved due to the elimination of outliers achieved by avoiding much of the turbulent areas in the wave front's wake.

A new merging point between the normal and OS is found by seeing where the shocks crosses after the OS template has been run a second time. The result is seen in Figure 3.17.

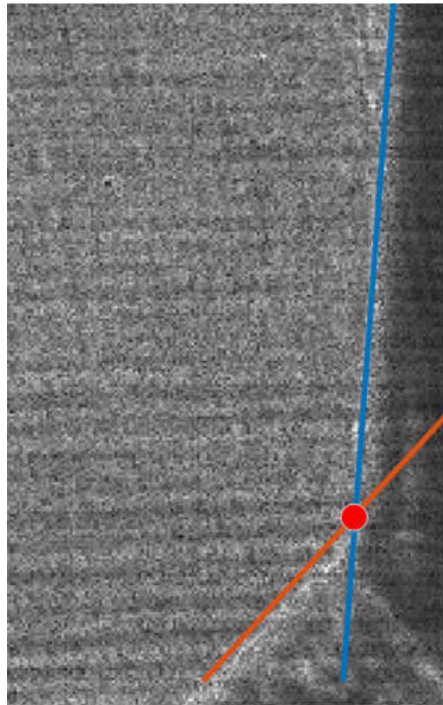


Figure 3.17: The result of the second run of the OS template. The red dot represents the new merging point. This figure represents the final step in the wave front tracking process.

The result shown is the final step in the wave front tracking process and a base is made for calculating angles and velocities.

### 3.7 Angle Estimation

By using 1<sup>st</sup> degree least square fit for estimating both NS and OS, i.e.  $y = ax + b$ , the slope of each shock is given by “ $a$ ”, and thereby the angle is calculated as shown in Formula (3.4).

$$\alpha = \frac{\pi}{2} \tan^{-1}(a) \quad (3.4)$$

Where

$a$  = the median slope of all the estimated shocks in a video

$\alpha$  = alpha, the angle of the shock

The method of calculating the shock angle, as shown in Formula (3.4), is applied to both shocks.

### 3.8 Velocity Estimation

The velocity is estimated by comparing the position of the wave front in the first wave front image with the last wave front image. How many pixels the wave front moves during a set of frames determines its velocity.

Additionally, the OS appears to move at a greater velocity than the NS. By applying the same formula for the OS as the NS, the velocities are calculated by using Formula (3.5).

$$v = \frac{\left( \Delta x \frac{P_{mm}}{px} \right)}{T_{cam} N_{frames}} \quad (3.5)$$

Where

$v$  = the velocity of the shockwave  $\left[ \frac{mm}{s} \right]$

$\Delta x$  = the number of pixels the shockwave has moved in total

$\frac{P_{mm}}{px}$  = pixel to millimeter conversion factor  $\left[ \frac{mm}{px} \right]$  (0.0775 or 0.105)

$T_{cam}$  = time elapsed per frame capture [s] ( $2e10^{-6}$  or  $5e10^{-6}$ )

$N_{frames}$  = the number of frames which are tracked

The reason for calculating wave front velocities globally, first to last frame, instead of locally, one frame to the next, is due to variations in the wave front estimate. Through visual inspection it comes clear that each wave front estimation might vary around  $\pm 7$  pixels while still being considered a satisfactory wave front estimate. Shown in Figure 3.18, the span of an accepted wave front estimate.

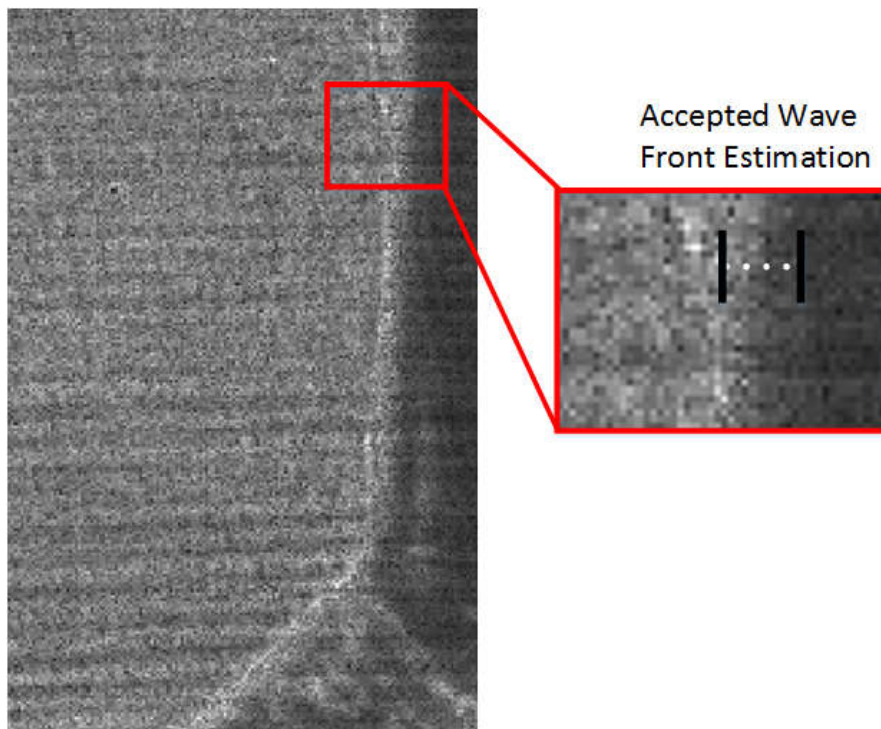


Figure 3.18: Illustration of expected wave front estimate. The black lines in the right-hand image displays the variation to be expected by a wave front estimate. The variation appears to be at around  $\pm 7$  pixels

If the velocity is to be calculated from one frame to the next,  $\pm 7$  pixels could make a significant impact on the velocity estimate as a wave front moves approximately 5 pixels per frame. However, calculating the velocity over 100 frames, using the first and last frame, reduces the impact of wave front estimate variance.

A different solution to estimating the velocity could be calculating the velocity from one frame to the next for all frames, then find the median velocity of all the estimated velocities. This is done and the result is shown and compared to the global velocity calculation in Chapter 4.1.



### 3.9 Triple Point Estimation

Estimating the triple shock is regarded a difficult endeavor. An assumption is made in consultation with CPSE, that the TP is estimated to reside slightly below the merging point. A formula is implemented marking a TP where the OS deviates a set number of pixels away from the NS. For the framework in its current state, the number of pixels the OS must deviate is 8. The TP is set using Formula (3.6).

$$TP = OS_i \{ |OS_i - NS_i| \geq 8, \quad \text{for } MP \leq i \leq N_{row} \quad (3.6)$$

Where

$TP$  = triple point

$OS_i$  = oblique shock at index  $i$

$NS_i$  = normal shock at index  $i$

$N_{row}$  = number of pixel rows in the image

$MP$  = breaking point

## 4 Results

The presented framework is tested on different high-speed videos in order to test the robustness of the framework. The results obtained through applying the framework to different high-speed videos are shown in Table 4.1 and Table 4.2.

Table 4.1: Velocity estimations of the shocks using proposed framework.

Experiment no.	Caption freq. <i>kfps</i>	Normal shock velocity $\left[\frac{m}{s}\right]$	Oblique shock velocity $\left[\frac{m}{s}\right]$	Triple point velocity	Conversion factor $\left[\frac{mm}{px}\right]$
2516	500	230	262	239	0.105
2533	200	471	473	473	0.105
2543	200	236	268	241	0.105
2573	500	220	261	232	0.0775
2592	500	450	486	460	0.0775

Table 4.2: Angle estimations of the shocks in the high-speed videos using proposed framework [10].

Experiment no.	Normal shock angle [ <i>deg</i> ]	Normal shock 95% confidence interval [ <i>deg</i> ]	Oblique shock angle [ <i>deg</i> ]	Oblique shock 95% confidence interval [ <i>deg</i> ]
2516	87	[86 ... 87]	50	[47 ... 53]
2533	89	[88 ... 91]	89	[88 ... 91]
2543	87	[86 ... 88]	54	[50 ... 58]
2573	87	[85 ... 87]	50	[45 ... 52]
2592	89	[85 ... 93]	62	[57 ... 63]

The chapters 4.2, 4.3, 4.4, 4.5 and 4.6 contain a short description of each experiment video presented in this thesis.

### 4.1 Global versus Local Velocity Estimation

As described in Chapter 3.8, another way of calculating the wave front velocities is to calculate the velocity of one frame to the next for all frames, then find the median. The result and comparison between global and local velocity estimation applied to the video in experiment 2592 is shown in Table 4.3 for NS and Table 4.4 for OS.

Table 4.3: Comparison of global and local velocity estimation for NS using experiment 2516 to compare the results.

Experiment No.	Local velocity NS $\left[\frac{m}{s}\right]$	Global velocity NS $\left[\frac{m}{s}\right]$	Deviation NS $\left[\frac{m}{s}\right]$
2516	235	230	5

Table 4.4: Comparison of global and local velocity estimation for OS using experiment 2516 to compare the results.

Experiment No.	Local velocity OS $\left[\frac{m}{s}\right]$	Global velocity OS $\left[\frac{m}{s}\right]$	Deviation OS $\left[\frac{m}{s}\right]$
2516	262	262	0

It is observed that both velocity estimations provide a feasible answer. As there is no way to compare the results to a definitive answer. It is informed by CPSE that the results provided has an unknown uncertainty, however,  $\pm 10 \left[\frac{m}{s}\right]$  uncertainty is not unlikely, though not documented.

Regarding the accuracy of global versus local velocity estimation, it becomes unclear as to which is the better solution. For this thesis, the velocity estimates presented are calculated globally.

## 4.2 Experiment No. 2516

Experiment no. 2516 consisted of a relatively good quality video compared to the other videos supplied. A typical frame from this experiment is seen in Figure 4.1.

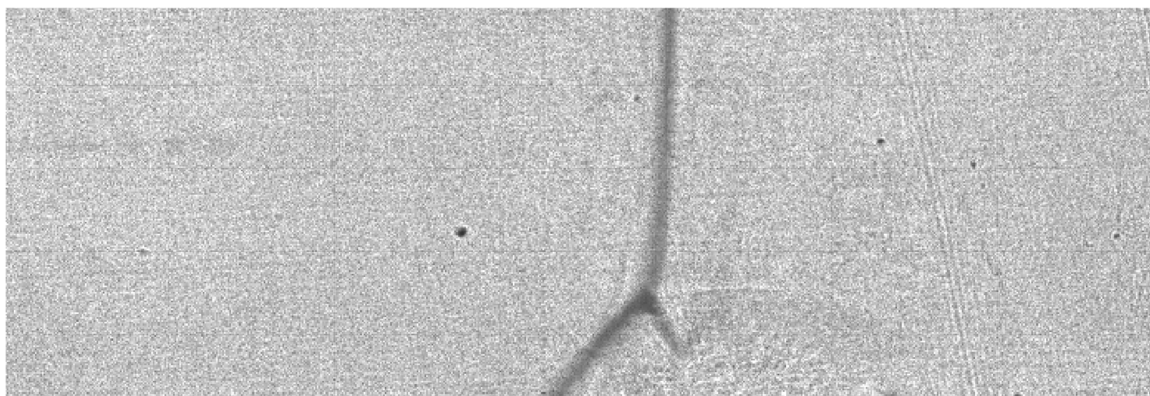


Figure 4.1: Typical frame from the 2516 experiment. Salt &amp; pepper noise can be seen throughout the image.

The total result of tracking all wave fronts in experiment 2516 can be seen in Figure 4.2. Note that a small gap is seen for every 10<sup>th</sup> frame. These were the outlier frames removed as described in Chapter 3.1.

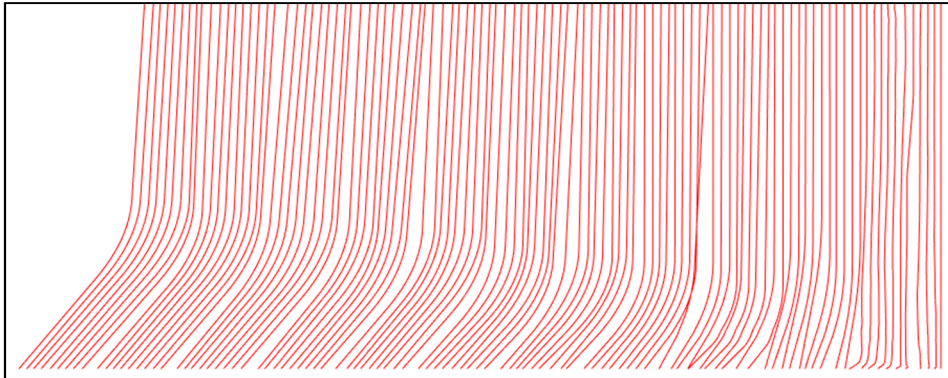


Figure 4.2: Result of all wave front estimations for experiment 2516 is displayed as the red lines. The wave front in the first frame is the rightmost red line, while the wave front in the last frame is the leftmost red line.

It is observed that the wave front appears to move close to the same distance in every frame, approximately 5 pixels.

### 4.3 Experiment No. 2533

Experiment no. 2533 is considered a relatively good quality video. However, the video stands out with its wave front velocity and non-observable OS. A typical frame of experiment 2533 is shown in Figure 4.3.

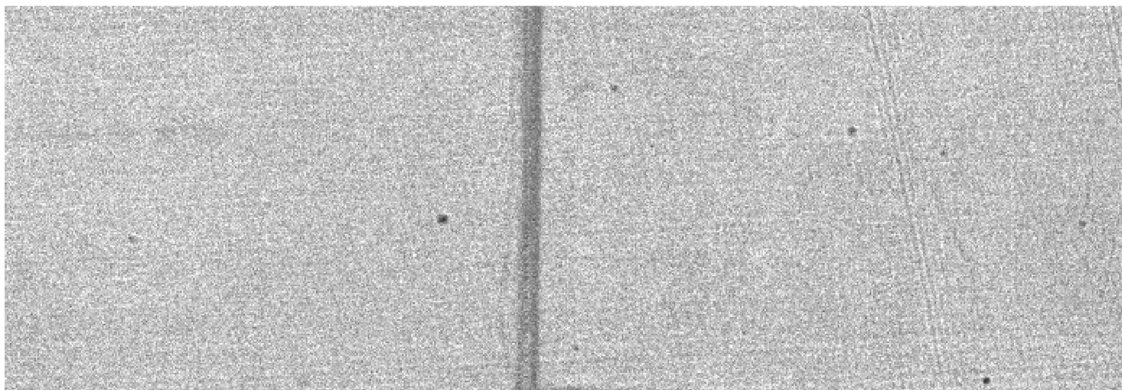


Figure 4.3: Typical frame from the 2533 experiment. Salt & pepper noise can be seen throughout the image.

The total result of tracking all wave fronts in experiment 2533 can be seen in Figure 4.4. Note that a small gap is seen for every 10<sup>th</sup> frame. These were the outlier frames removed as described in Chapter 3.1.

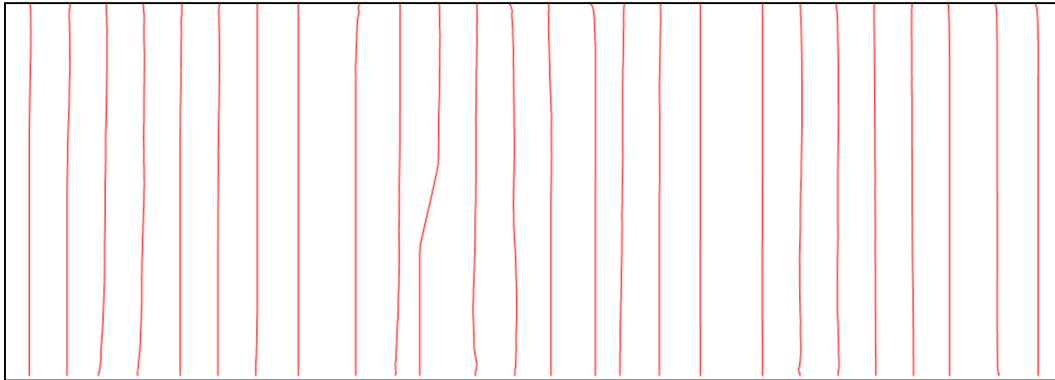


Figure 4.4: Result of all wave front estimations for experiment 2533 is displayed as the red lines. The wave front in the first frame is the rightmost red line, while the wave front in the last frame is the leftmost red line.

It is observed that the wave front appears to move close to the same distance in every frame, approximately 10 pixels.

#### 4.4 Experiment No. 2543

Experiment no. 2543 is considered a relatively high quality video. However, the video is obstructed by a pressure sensor wire. The pressure sensor wire does in no way influence the wave front, but it does influence the framework which uses the images as inputs. A typical frame of experiment 2543 is shown in Figure 4.5.

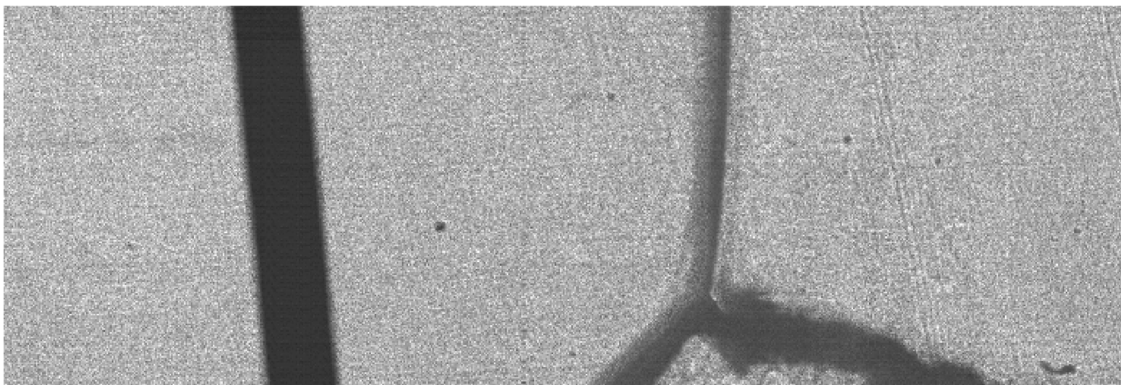


Figure 4.5: A typical frame from the 2543 experiment. A pressure sensor wire seen as the distinct black line in the image's left-hand side. Additionally, salt & pepper noise can be seen throughout the image.

The total result of tracking all wave fronts in experiment 2543 can be seen in Figure 4.6. Note that a small gap is seen for every 10<sup>th</sup> frame. These were the outlier frames removed as described in Chapter 3.1.

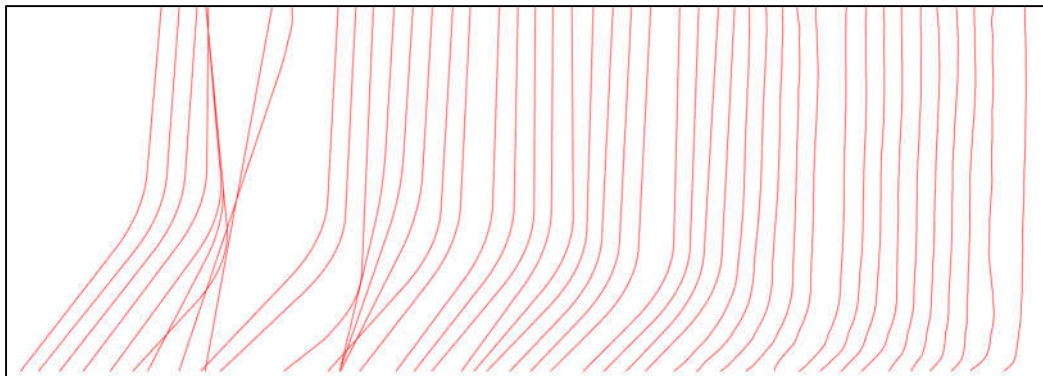


Figure 4.6: Result of all wave front estimations for experiment 2543 is displayed as the red lines. The wave front in the first frame is the rightmost red line, while the wave front in the last frame is the leftmost red line.

It is seen by the results in Figure 4.6, how the obstructing pressure sensor wire influences the framework results. It is also observed that all wave front estimations are in accordance with expected results, except for the images with the wave front in close proximity to the obstructing sensor wire. Despite having an obstruction in the video, enough wave fronts were accurately tracked to get satisfactory angle, velocity and TP estimates.

## 4.5 Experiment No. 2573

Experiment no. 2573 was considered a medium quality video and was chosen as the first video to base the framework on, before applying the framework to other videos. The video contained more interference compared to the other experiments mentioned in Chapter 4.2, 4.3, 4.4 and 4.6, but in turn, the wave front was more apparent. A typical frame from the 2573 experiment is shown in Figure 4.7.

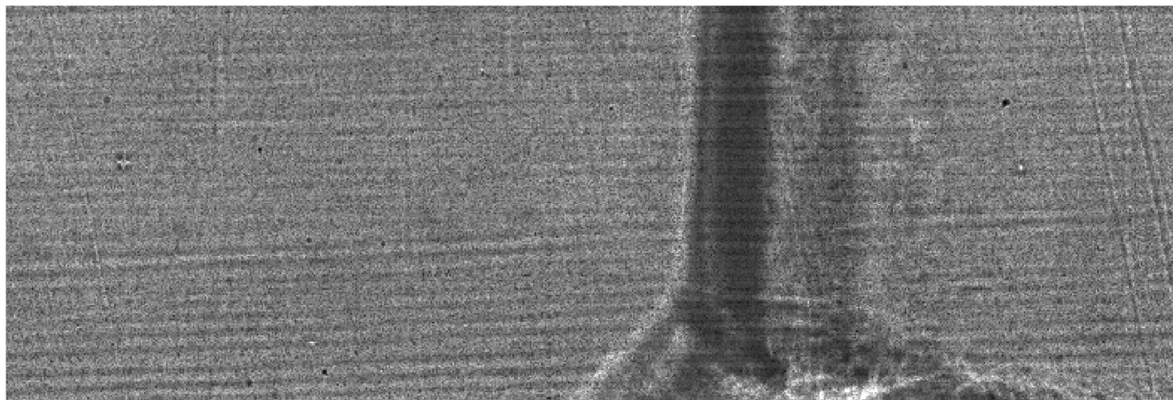


Figure 4.7: A typical frame from the 2573 experiment. The interference mentioned which is prominent in this experiment aims at the horizontal and vertical “ridge”-lines which appear in the entire frame. Additionally, Gaussian noise can be seen throughout the image.

The total result of tracking all wave fronts in experiment 2573 can be seen in Figure 4.8. Note that a small gap is seen for every 10<sup>th</sup> frame. These were the outlier frames removed as described in Chapter 3.1.

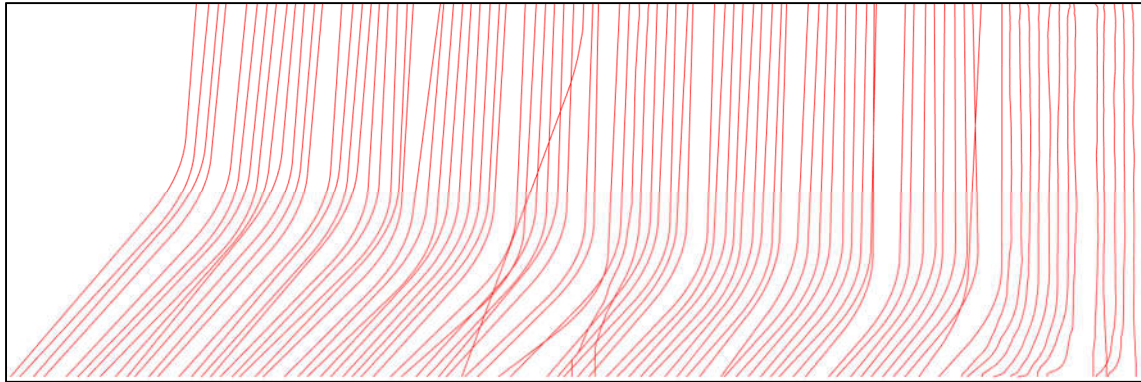


Figure 4.8: Result of all wave front estimations for experiment 2573 is displayed as the red lines. The wave front in the first frame is the rightmost red line, while the wave front in the last frame is the leftmost red line.

A prominent outlier wave front is observed in the wave front result plot. In addition, it is seen that the framework found it more challenging to estimate the wave fronts of this experiment compared to experiment 2516 as described in Chapter 4.2. This is due to video quality variations of the different experiments.

## 4.6 Experiment No. 2592

Experiment no. 2543 is considered a relatively poor quality video. The interference is prominent while the wave front is vague. A typical frame from experiment 2592 can be seen in Figure 4.9.

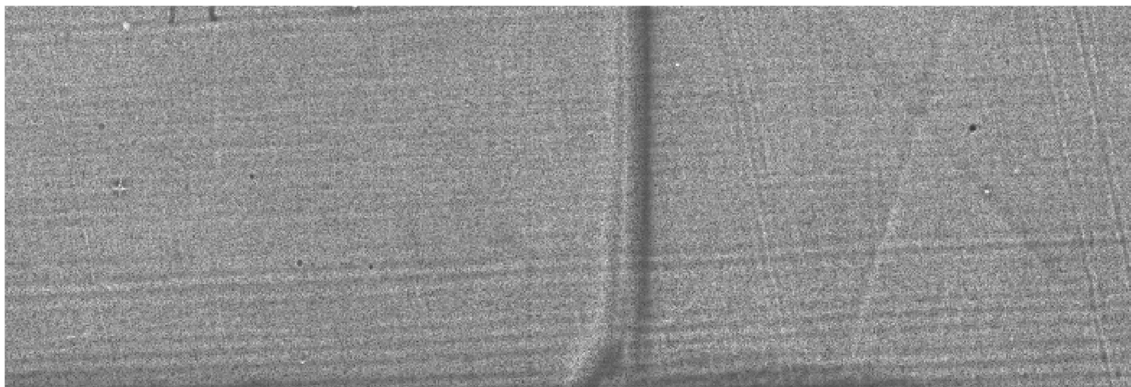


Figure 4.9: Typical frame from experiment 2592. Interference is somewhat prominent while the wave front is vague.

The total result of tracking all wave fronts in experiment 2592 can be seen in Figure 4.10. Note that a small gap is seen for every 10<sup>th</sup> frame. These were the outlier frames removed as described in Chapter 3.1.

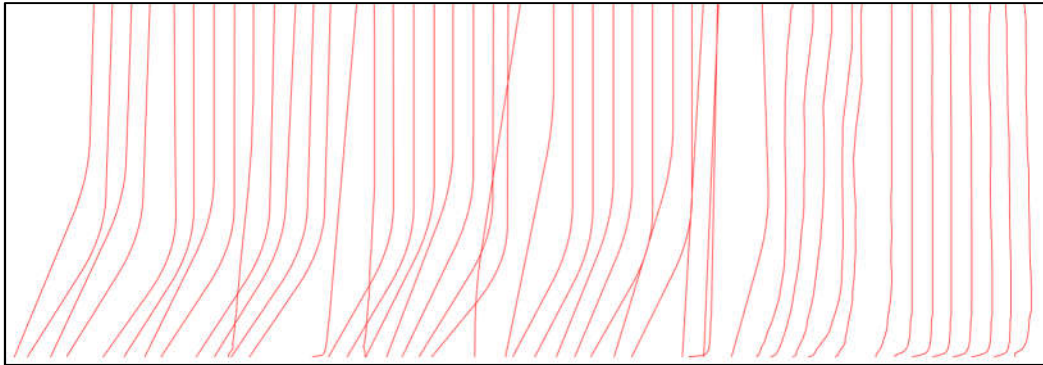


Figure 4.10: Result of all wave front estimations for experiment 2592 is displayed as the red lines. The wave front in the first frame is the rightmost red line, while the wave front in the last frame is the leftmost red line.

This had a particularly vague wave front, however, enough wave fronts were accurately tracked to get satisfactory angle and velocity estimates.

## 4.7 Result Comparison with CPSE

For comparison, the CPSE has supplied velocity calculations based on pressure sensor measurements as shown and compared to the proposed framework in Table 4.5.

Table 4.5: Comparison of velocity results between proposed framework and calculations done by the Combustion, Process Safety and Explosions research group at USN. The research group velocities are calculated using pressure sensor data. All velocities are of the OS.

Experiment no.	Research group velocity $\left[\frac{m}{s}\right]$	Framework velocity $\left[\frac{m}{s}\right]$	Deviation $\left[\frac{m}{s}\right]$
2516	266	262	4
2533	485	473	12
2543	264	268	4
2573	267	261	6

Experiment no. 2533 is more challenging for the framework to estimate as the OS angle is close to identical to the NS angle. This means that the NS spanned close to the entire wave front as shown in Figure 4.11.



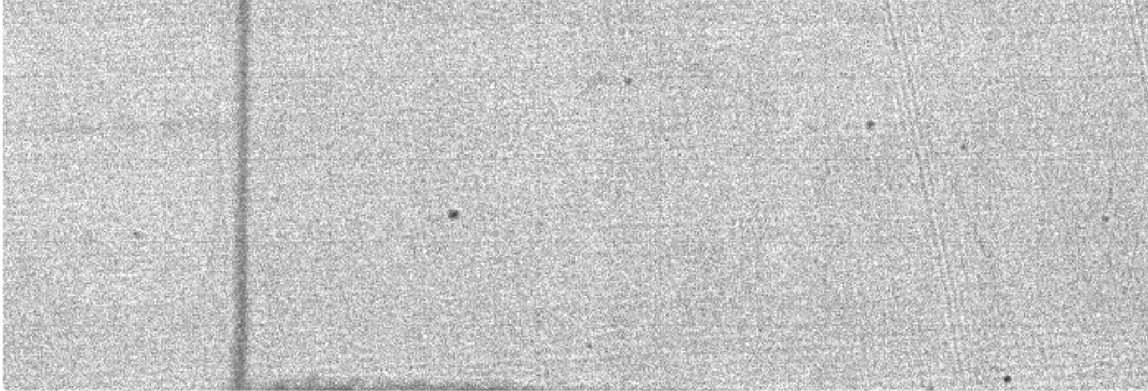


Figure 4.11: Raw image of the wave front in experiment no. 2533 with the OS being close to negligible. Meaning OS angle and velocity are almost identical to that of the NS.

## 5 Discussions

This chapter addresses limitations, modifications and other factors which could reduce or increase performance of the proposed framework.

### 5.1 Normalization of Data

The image data is not normalized when introduced to the framework which would be preferred.

### 5.2 Outliers

The outlier removal presented in Chapter 3.2.3 is somewhat trivial and most likely does not present the best option available in this regard. Complex outlier removal techniques were regarded as out of scope for this thesis, as outlier removal could very well be a Master's Thesis by its own.

### 5.3 Effectiveness and Accuracy of Framework

The effectiveness of the framework is restricted to some degree by the competence of the user. As the user is required to adjust the template values as well as selecting starting and ending frame of the wave front sequence and marking the wave front in each frame. However, the framework is still regarded as easy-to-use, as the actions required of the user are solved by only clicking the mouse on the frames which are displayed.

A different factor which influences the framework effectiveness is the quality of the video it is applied to.

### 5.4 Error Margins

As mentioned in Chapter 3.8, an approved wave front estimate is expected to vary about +/-7 pixels. The impact of the wave front estimate variation becomes significant due to the time between each frame capture being relatively short, i.e.  $2e-10[s]$ . In a worst-case scenario calculating local velocity estimations, if the first wave front appears at -7 pixels and the following wave front appears at +7 pixels, using the distance and framerate conversion factor from experiment 2516 and applying Formula (3.5) the result is shown in Formula (5.1).

$$v = \frac{\left( \Delta x \frac{P_{mm}}{px} \right)}{T_{cam} N_{frames}} = \frac{(7 + 5 + 7) 0.105}{0.000002 1} = 997500 \left[ \frac{mm}{s} \right] \approx 998 \left[ \frac{m}{s} \right] \quad (5.1)$$

An assumption, using Formula (5.1) is made, which is that the wave front moved on average 5 pixels per frame throughout the video. The assumption is made based on visual inspection of result plots. A comparison between the expected result and the worst-case scenario regarding experiment 2516 is seen in Table 5.1.

Table 5.1: Comparison of velocity estimation using expected result from global velocity estimation and worst-case scenario with local velocity estimation.

Experiment no.	Expected result $\left[\frac{m}{s}\right]$	Worst case $\left[\frac{m}{s}\right]$	Deviation $\left[\frac{m}{s}\right]$
2516	230	998	768

An alternative way of calculating the wave front velocities locally throughout the video is shown in Chapter 4.1 and may prove to be as good a method as the global velocity estimation presented in this thesis.

## 5.5 Template Angle Acceptance

An assumption is made that the wave fronts appear at about 90 and 45 degrees. The further a wave front deviates from these assumptions, the less accurate the framework will be. The templates used in the proposed method search distinctively for wave fronts which are 90 and 45 degrees. However, the width and height of the template matrices allow for some leeway regarding the angle of the wave fronts.

Had the wave fronts appeared at different angles than those observed in the high-speed videos, the framework templates would be constructed differently. However, in the proposed framework, changing the templates to achieve different optimal angle matching is achievable by changing the skew of the templates.

Using the experiment no. 2516 video, a test is made for angle acceptance of the template matching. The video frames are first rotated 10 degrees counterclockwise, then template matching is then run throughout the video. A result of the test is seen in Figure 5.1.

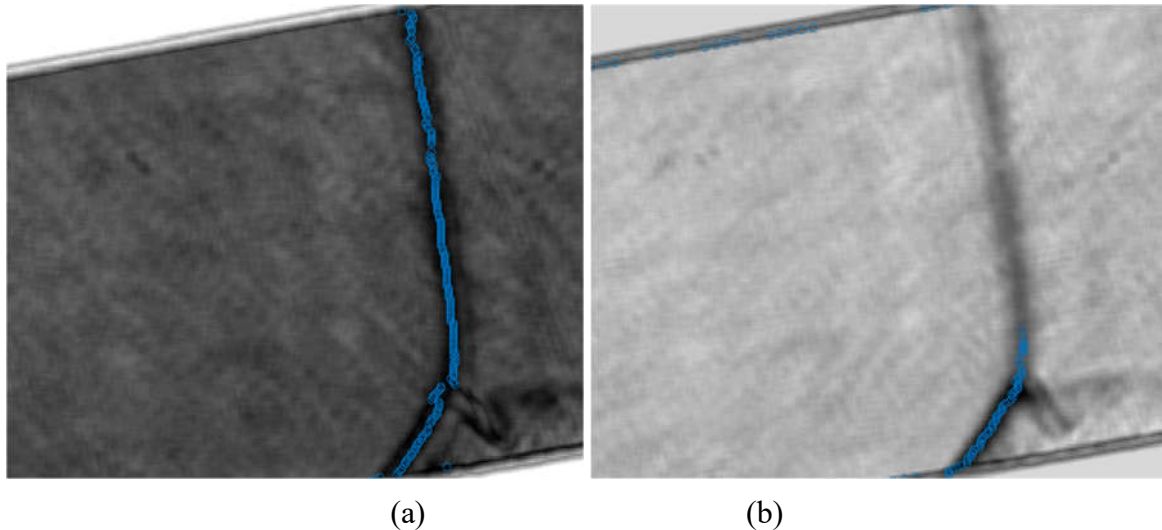


Figure 5.1: (a) and (b) shows the NS and OS template matching result, respectively.

The results show promise for the performed test. The raw image of which the templates are applied to is seen in Figure 5.2.

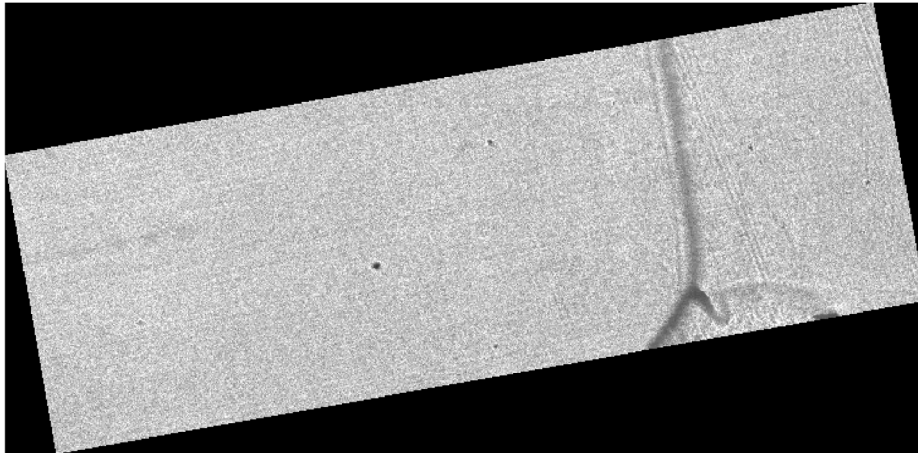


Figure 5.2: Raw image of the counterclockwise rotated frame which the NS and OS templates were applied to. The second test includes rotating the frame 10 degrees in the clockwise direction, instead of counterclockwise. The results are seen in Figure 5.3.

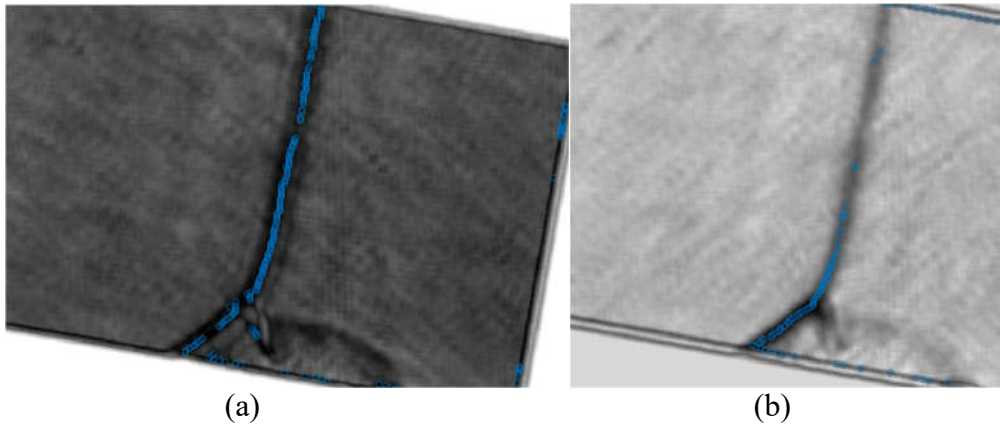


Figure 5.3: Result of NS and OS templates applied to a 10 degrees clockwise rotated wave front image. The raw 10 degree clockwise rotated image is seen in Figure 5.4.

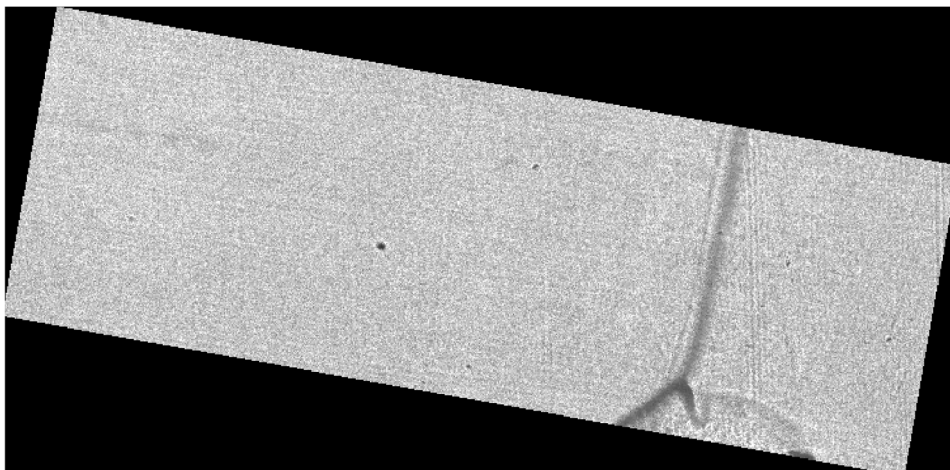


Figure 5.4: Raw image of the clockwise rotated frame which the NS and OS templates were applied to. Note that the results presented in the described tests are achieved without the sliding window technique which would further improve wave front tracking and reduce outliers.

## 6 Conclusions

The task of developing a wave front tracking framework for the Combustion, Process Safety and Explosions research group at USN was approached with focus on finding a viable solution. As the high-speed videos provided is blurred and corrupted with noise, developing a solution which shows promise was prioritized. The result is a framework that estimates normal and oblique shock angle and velocity, as well as triple point estimation in each frame and triple point velocity in the given video. The framework is limited by the quality of the video provided, as the video is the only source of information the framework receives. Certain inputs from the user is required in order to ensure optimal performance of the framework.

The framework developed was first implemented on a single high quality frame as a way to do a feasibility test during development. Secondly, the framework was expanded to automatically run for an entire video. Lastly, the framework was tested on different high-speed videos in order to test diversity of the framework.

An extended Segmented Regression method was developed in order to handle two separate data sets in parallel.

The proposed framework delivers results in accordance with theoretically calculated values and measured data presented by the Combustion, Process Safety and Explosions research group at USN and Caltech.

### 6.1 Future Work

This chapter mentions work that would increase performance and accuracy of the proposed framework.

#### 6.1.1 Code Optimization

The code should be optimized to increase speed of the framework. Additionally, the code should be generalized, making it easier to implement on videos with different resolutions and framerates.

#### 6.1.2 Outlier Removal

As mentioned, the outlier removal in this thesis is trivial compared to other complex techniques. A robust outlier removal technique would increase framework accuracy and further reduce outlier interference.

#### 6.1.3 Normalization of Data

The framework presented in this thesis does not operate using normalized data as would be preferred. Normalizing data would make the framework more generalized and easier implemented using other videos with different pixel values.

### 6.1.4 Template Value Update

Having correct template values for the template matching is crucial to obtain a satisfactory result using the proposed framework. By implementing a method which allows the for dynamic template values is to be preferred. However, one major challenge in obtaining correct updated template values is where on the image the template values should be selected.

One solution could be using the Otsu threshold method to obtain a rough estimate of where the wave front exists in each frame. As described in Chapter 2.1, the Otsu method finds a threshold attempting to segment objects from the background. A result of this is seen in Figure 6.1.

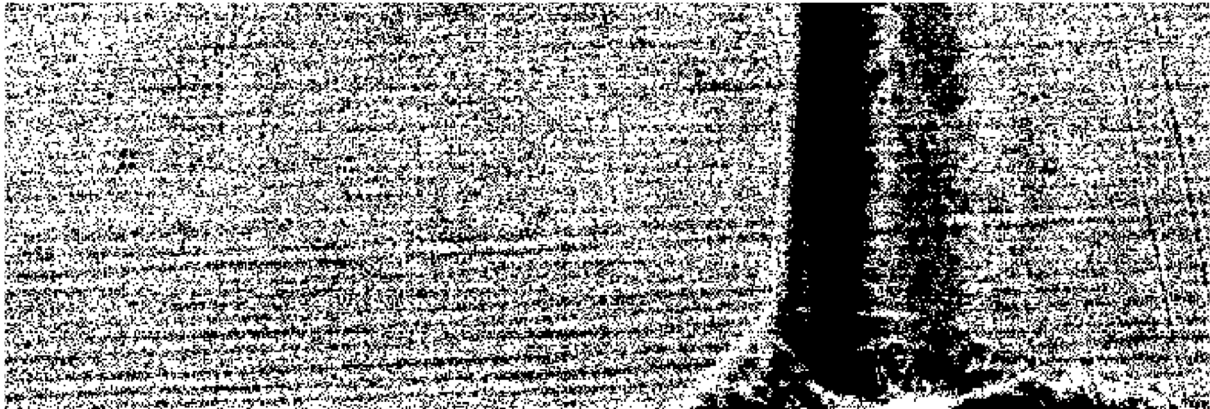


Figure 6.1: Output image of the Otsu threshold technique.

Using the image seen in Figure 6.1, it becomes easier to see where the wave front exists in the image. Coordinates can be extracted automatically from the image generated by the Otsu method, and the coordinates will tell the framework where to extract new template values.

### 6.1.5 Oblique Shock Template Rework

Re-constructing the OS template might improve performance as the proposed OS template does not maximize its OS overlap. The proposed framework is seen in Figure 6.2.

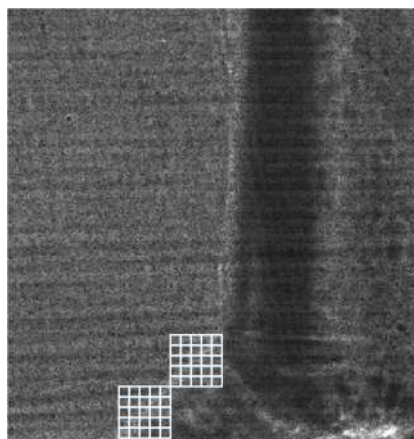


Figure 6.2: Layout of OS template as proposed in this thesis. Template not to scale.

A re-constructed template could be constructed as seen in Figure 6.3.

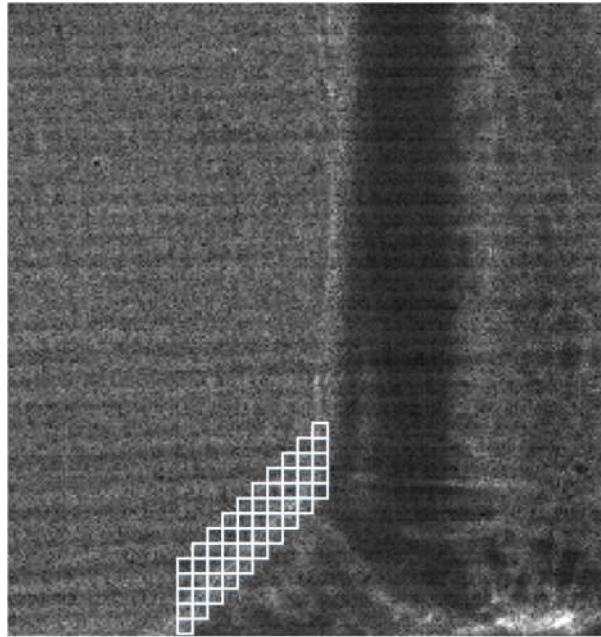


Figure 6.3: Proposal of re-constructed OS template. Template not to scale.

The proposed reconstructed OS template would have a larger OS overlap and in turn might achieve a better match during the template matching procedure described in Chapter 3.2.2.

# References

- [1] N. Otsu, “*A Threshold Selection Method from Gray-Level Histograms*”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.
- [2] MathWorks. (2016). “*Graythresh*”. (Accessed 05.02.2017) URL: <https://www.mathworks.com/help/releases/R2016b/images/ref/graythresh.html>
- [3] S. Bhardwaj, A. Mittal (2012). “*A Survey on Various Edge Detector Techniques*”. (Accessed 10.03.2017) URL: [http://ac.els-cdn.com/S221201731200312X/1-s2.0-S221201731200312X-main.pdf?\\_tid=2e66faee-311c-11e7-9ff2-00000aab0f01&acdnat=1493938456\\_f3c2fa0b010f920c87f9a17a81b5252e](http://ac.els-cdn.com/S221201731200312X/1-s2.0-S221201731200312X-main.pdf?_tid=2e66faee-311c-11e7-9ff2-00000aab0f01&acdnat=1493938456_f3c2fa0b010f920c87f9a17a81b5252e)
- [4] Adaptive-Vision (2017). “*Template Matching*”. (Accessed 13.02.2017) URL: [http://docs.adaptive-vision.com/4.7/studio/machine\\_vision\\_guide/TemplateMatching.html](http://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html)
- [5] A. Coste (2012). “*Filtering, Edge detection and Template matching* “. (Accessed 15.02.2017) URL: [http://www.sci.utah.edu/~acoste/uou/Image/project2/ArthurCOSTE\\_Project2.pdf](http://www.sci.utah.edu/~acoste/uou/Image/project2/ArthurCOSTE_Project2.pdf)
- [6] A. Solberg (2016). “*INF 4300 – Digital Image Analysis*”. (Accessed 22.02.2017) URL: <http://www.uio.no/studier/emner/matnat/ifi/INF4300/h16/undervisningsmateriale/inf4300-2016-f04-segmentation.pdf>
- [7] R. Fisher, S. Perkins, A. Walker, E. Wolfart (2000). “*Fourier Transform*”. (Accessed 24.03.2017) URL: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm>
- [8] R.J. Oosterbaan(1994). “*Regression Analysis*”. (Accessed 22.03.2017) URL: <https://www.waterlog.info/pdf/regtxt.pdf>
- [9] Wikipedia (January 2017), “*Segmented Regression*”. (Accessed 12.04.2017) URL: [https://en.wikipedia.org/wiki/Segmented\\_regression](https://en.wikipedia.org/wiki/Segmented_regression)
- [10] G. G. Løvås, “*Statistikk for Universiteter og Høgskoler*”. 2. edition. Oslo: Universitetsforlaget, 2012.



# Appendices

Appendix A Task Description

Appendix B SIMS Abstract – Estimation of Shock Waves Velocity based on Image Processing and Optimization

Appendix C MATLAB Code