Journal of Signal and Information Processing, 2013, 4, 109-113
doi:10.4236/jsip.2013.43B019 Published Online August 2013 (http://www.scirp.org/journal/jsip)

Image Processing Techniques in Shockwave Detection and Modeling

Suxia Cui[1], Yonghui Wang[2], Xiaoqing Qian[3], Zhengtao Deng[3]

[1]Department of Electrical and Computer Engineering, Prairie View A&M University, Prairie View, Texas, USA; [2]Department of Engineering Technology, Prairie View A&M University, Prairie View, Texas, USA; [3]Department of Mechanical Engineering, Alabama A&M University, Huntsville, Alabama, USA.
Email: sucui@pvamu.edu, yowang@pvamu.edu, xiaoqing.qian@aamu.edu, zhengtao.deng@aamu.edu

Received May, 2013.

ABSTRACT

Shockwave detection is critical in analyzing shockwave structure and location. High speed video imaging systems are commonly used to obtain image frames during shockwave control experiments. Image edge detection algorithms become natural choices in detecting shockwaves. In this paper, a computer software system designed for shockwave detection is introduced. Different image edge detection algorithms, including Roberts, Prewitt, Sobel, Canny, and Laplacian of Gaussian, are implemented and can be chosen by the users to easily and accurately detect the shockwaves. Experimental results show that the system meets the design requirements and can accurately detect shockwave for further analysis and applications.

Keywords: Shock Wave Detection; Image Edge Detection

1. Introduction

A shockwave is a strong compression wave existing in supersonic/hypersonic flow field. Across the shockwave gas pressure, temperature and density increases significantly as a function of flow Mach number. Near the nose area of a supersonic/hypersonic flight vehicle, strong shockwaves exist. Extreme heat fluxes and heat load to the vehicle surface requires strong thermal protection in the nose area. Additionally, the shock standoff distance varies drastically with the temperature for a non-ideal gas, causing large differences in the heat transfer to the thermal protection system and drag of the vehicle. Supersonic wind tunnel experiments are needed to provide insight into the physics of air flow at high Mach numbers. Knowing the location and shape of the shockwave is essential to the success of vehicle design. In general, optical imaging system can be used to capture shockwave shape, location and flow property changes near the test article in a supersonic wind tunnel. High speed video imaging system and Schlieren system will normally generate significant amount of images that need to be analyzed after the experiment. Image processing techniques can make this type of data analysis more efficient and precise, but there exists challenges on how to pre-process the obtained images according to the wind tunnel condition as well as dealing with large volume of data cap tured by high speed camera. Data acquisition is performed in wind tunnel, while data analysis will be completed by a powerful PC.

2. Shockwave Stand-off Distance Detection

A series of shockwave control experiments were conducted using a supersonic wind tunnel facility. The supersonic flow is created by a primary supersonic nozzle to provide supersonic speed up to Mach 4. The flow emerging from the nozzle is then exhausted as a free jet into a windowed low pressure test cabin with extremely low back pressure. **Figure 1** shows the shockwave experiment setting and a typical experimental image captured by a high speed video camera for a Mach 4 shockwave control experiment. The test article basic diameter (perpendicular to the incoming flow direction) is 3cm and the nose is spherical. The length of the cylindrical part is around 7 cm with the total length of the model from nose tip to cylinder base is 10cm. The shockwave structure and location capturing device were captured by a high speed video imaging system with speed of 2000 frames per second at 512 x 512 resolutions. Around 8000 frames of image data was collected for each experiment.

To analyze the shockwave stand-off distance, a special shockwave shape and stand-off distance detection algorithm needs to be developed. This algorithm should be able to accurately capture the steady shape and unsteady shape change of the shockwave especially the shockwave stand-off distance along the stagnation line. Since the experimental image has built-in noise, a noise cancellation algorithm should be incorporated in the detection scheme. The detection algorithm should also be robust because the large amount of unsteady shockwave images has to be analyzed. The time history of the shockwave stand-off distance under control conditions has to be precisely captured.

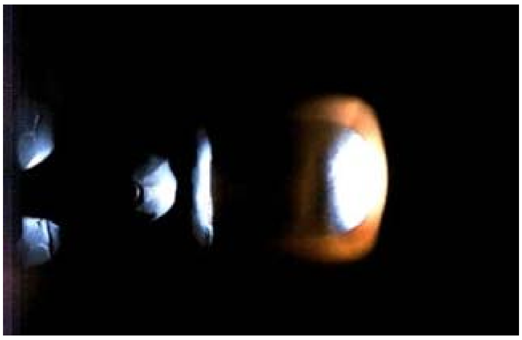


Figure 1. Aero-Test Article inside the wind tunnel and typical shockwave image near the nose are of the test article.

The current research focused on the development of a new shockwave standoff distance detection algorithm. To detect the shockwave shape and stand-off distance through the images, image processing techniques can be used to make this type of data analysis more efficient and precise. In the current research, a new interactive and user-friendly shockwave detection software was developed. This software can precisely detect the shockwave stand-off distance in front of the test article inside a supersonic wind tunnel.

3. Image Edge Detection Algorithms

Edge detection algorithm is the natural choice for detecting shockwave within the obtained high speed video sequences. Edges are local features of images, which are local regions with special properties. Edges are pixels with significant local changes of intensity–large gradient–in an image. Edges typically occur on the boundary between two different regions in an image (as shown in **Figure 2**). Because important features, such as corners, lines, curves, etc., can be extracted from the edges, edge detection algorithms are very important in computer vision applications. Edge detection is to detect the edges and to produce a line drawing of a scene from an image of that scene.

There are four steps for edge detection: (1) Smoothing: suppress as much noise as possible, without destroying the true edges. (2) Enhancement: apply a filter to enhance the quality of the edges in the image (sharpening). (3) Detection: determine which edge pixels should be discarded as noise and which should be retained (usually, thresholding provides the criterion used for detection). (4) Localization: determine the exact location of an edge Edge thinning and linking are usually required in this step.

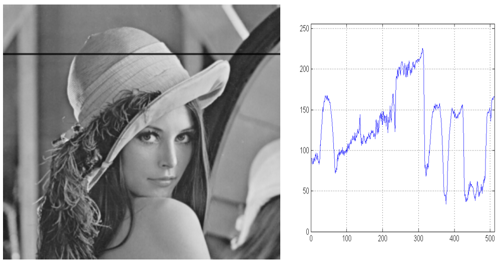


Figure 2. Image edge example: a scan line highlighted (left) and intensity along the highlighted scan line (right).

In mathematics, derivatives are used to describe changes of continuous functions. Applying this to 2D images, partial derivatives can be used to express the image intensity changes (e.g. edges). Edge points can be detected by detecting (1) the local maxima or minima of the first derivative and (2) the zero-crossing of the second derivative. For 1D discrete signals, the first derivative can be approximated as

$$f'(x) = \lim_{a \to 0} \frac{f(x+a) - f(x)}{a} \approx f(x+1) - f(x);$$

and the second derivative can be approximated as

$$\begin{aligned} f''(x) &= \lim_{a \to 0} \frac{f'(x) - f'(x-a)}{a} \\ &\approx f'(x) - f'(x-1) \\ &\approx f(x+1) - 2f(x) + f(x-1). \end{aligned}$$

For 2D images, gradient can be used to describe edges. Gradient is defined as a vector

$$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}],$$

its magnitude,

$$\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2},$$

provides the edge's strength information. For images, by using pixel-coordinate notation (e.g., i for row number and j for column number, and I for image signal), the gradient can be approximated as

$$\begin{aligned} \frac{\partial I}{\partial x} &= \lim_{a \to 0} \frac{I(x+a, y) - I(x, y)}{a} \approx I(x+1, y) - I(x, y) \\ &= I(i, j+1) - I(i, j), \text{ and} \\ \frac{\partial I}{\partial y} &= \lim_{a \to 0} \frac{I(x, y+a) - I(x, y)}{a} \approx I(x, y+1) - I(x, y) \\ &= I(i+1, j) - I(i, j). \end{aligned}$$

3.1. Roberts

The Roberts method finds edges using the Roberts approximation to the derivative. It returns edges at those points where the gradient of I is maximum. It was one of the first edge detectors and was initially proposed by Lawrence Roberts [1]. The Roberts edge detector approximates the partial derivatives of the gradient as

$$\frac{\partial I}{\partial x} \approx I\left(i, j\right) - I\left(i+1, j+1\right),$$

$$\frac{\partial I}{\partial y} \approx I\left(i+1, j\right) - I\left(i, j+1\right).$$

This approximation can be implemented by convolving the following masks

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } M_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

onto images. An example of Roberts edge detection is shown in **Figure 3**.

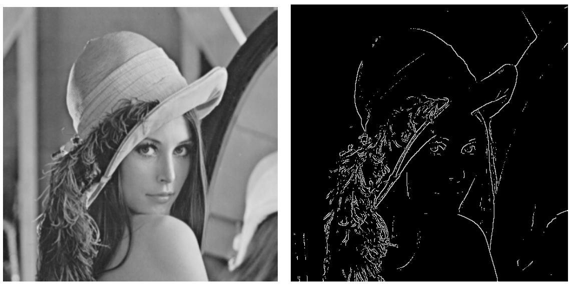


Figure 3. Roberts edge detector: original image (left) and detected edges (right).

3.2. Prewitt

The Prewitt method finds edges using the Prewitt approximation to the derivative. It was proposed by Judith M. S. Prewitt [2]. The Prewitt operator approximates the partial derivatives of the gradient as

$$\frac{\partial I}{\partial x} \approx I\left(i-1, j+1\right) + I\left(i, j+1\right) + I\left(i+1, j+1\right)$$
$$-I\left(i-1, j-1\right) - I(i, j-1) - I(i+1, j-1),$$

$$\frac{\partial I}{\partial y} \approx I\left(i+1, j-1\right) + I\left(i+1, j\right) + I\left(i+1, j+1\right)$$
$$-I\left(i-1, j-1\right) - I(i-1, j) - I(i-1, j+1).$$

The approximation can be implemented by convolving the following masks

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

onto images. An example of Prewitt edge detection is shown in **Figure 4**.

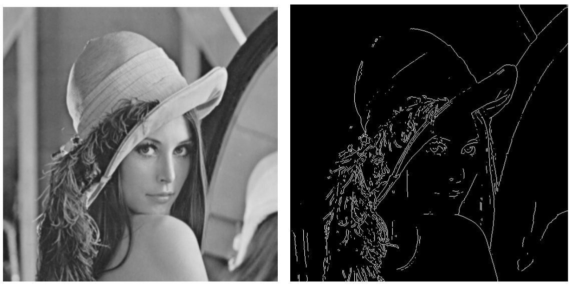


Figure 4. Prewitt edge detector: original image (left) and detected edges (right).

3.3. Sobel

The Sobel method finds edges using the Sobel approximation to the derivative. It was proposed by Irwin E. Sobel in 1970 [3]. A little bit different from the Prewitt operator, the Sobel operator approximates the partial derivatives of the gradient as

$$\frac{\partial I}{\partial x} \approx I\left(i-1, j+1\right) + 2I\left(i, j+1\right) + I\left(i+1, j+1\right)$$
$$-I\left(i-1, j-1\right) - 2I(i, j-1) - I(i+1, j-1),$$

$$\frac{\partial I}{\partial y} \approx I\left(i+1, j-1\right) + 2I\left(i+1, j\right) + I\left(i+1, j+1\right)$$
$$-I\left(i-1, j-1\right) - 2I(i-1, j) - I(i-1, j+1).$$

The approximation can be implemented by convolving the following masks

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

onto images. An example of Sobel edge detection is shown in **Figure 5**.

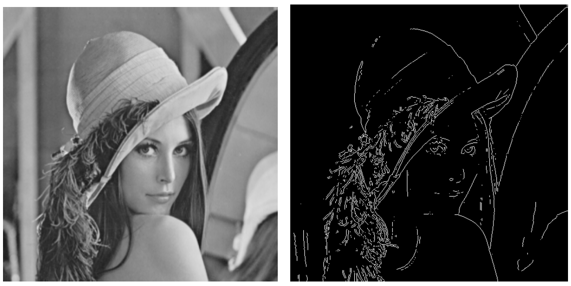


Figure 5. Sobel edge detector: original image (left) and detected edges (right).

Edge detection algorithms with Roberts, Prewitt, and Sobel operators share the same procedure in detecting edges. Main steps in edge detection with these operators: (1) calculate partial derivatives of the image through convolution, $I_x = \frac{\partial I}{\partial x} = I * M_x$ and $I_y = \frac{\partial I}{\partial y} = I * M_y$; (2) calculate the magnitude of gradient,

$\|\nabla I\| = \sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}$; and (3) if $\|\nabla I(i,j)\| > T$, then it is possible an edge point, T is the threshold.

3.4. Canny

Also being a method which finds edges by looking for local maxima of the gradient of the image, the Canny algorithm, which was developed by John F. Canny in 1986 [4], introduces more steps to minimize the noise affection. To reduce the noise effect, a low-pass Gaussian filter,

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} , \tag{1}$$

is applied to the image. By finding the derivative of the Gaussian filter, the gradient can be calculated

$$I_x = \frac{\partial I}{\partial x} = \frac{\partial}{\partial x}(I*G) = I*\frac{\partial}{\partial x}G = I*G_x ,$$

$$I_y = \frac{\partial I}{\partial x} = \frac{\partial}{\partial x}(I*G) = I*\frac{\partial}{\partial x}G = I*G_x ,$$

where $G_x = \frac{-x}{\sigma^2}G(x,y)$ and $G_y = \frac{-y}{\sigma^2}G(x,y)$.

After the gradient magnitude, $\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$, is obtained, two more steps are performed to identify the edges. A non-maxima suppression algorithm is applied to find the local maxima of the gradient magnitude, and a hysteresis thresholding with two thresholds is applied to get rid of extra noise and retain the real edge points. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges. An example of Canny edge detection is shown in **Figure 6**.

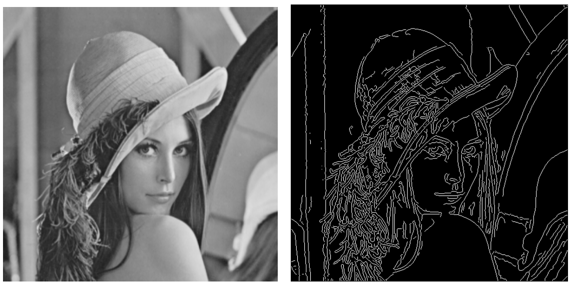


Figure 6. Canny edge detector: original image (left) and detected edges (right).

3.5. Laplacian of Gaussian

The Laplacian of Gaussian (LoG) method identifies edges by looking for zero crossings after filtering the image with an LoG filter [5].

In this method, a Gaussian low-pass filter as shown in Equation (1) is applied to reduce the noise effect. Previous discussion shows that edge points can be detected by detecting the zero-crossings of the second derivative. Here the Laplacian operator,

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

is applied to find the corresponding second derivatives and ultimately the zero-crossings. It can be shown that $\nabla^2(I*G) = I*\nabla^2 G$ and

$$\nabla^2 G(x,y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} = \frac{x^2+y^2-2\sigma^2}{\sigma^4}G(x,y)$$

Theoretically, the LoG function $g(x,y) = \nabla^2 G(x,y)$ has infinite extent; for a practical implementation of LoG edge detection, however, it has to be truncated to a finite size. It can be shown that the width of the center lobe of $g(x,y)$ is $w = 2\sqrt{2}\sigma$, and that the function

$$\hat{g}(x,y) = \begin{cases} g(x,y), & |x|,|y| \le \frac{3w}{2} \\ 0, & else \end{cases}$$

possesses around 99.7% of the energy of $g(x,y)$. By convolving the image with the appropriate LoG mask then detecting zero crossings in the convolution output, LoG method offers better localization. An example of LoG edge detection is shown in **Figure 7**.

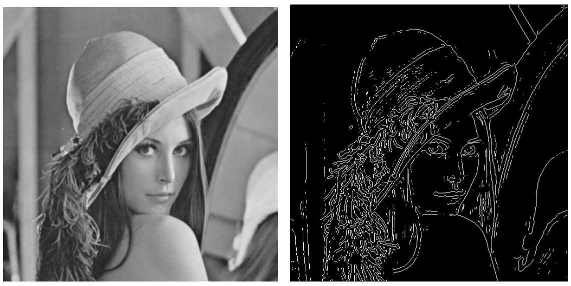
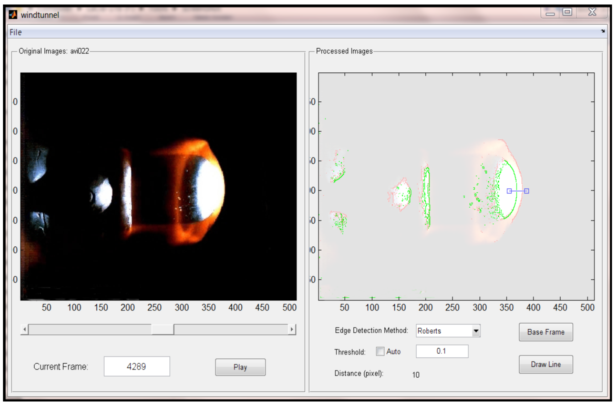


Figure 7. LoG edge detector: original image (left) and detected edges (right).

4. Shockwave Detection System and Results

To detect shockwave accurately and effectively, a MATLAB-based image processing software is developed. A user interface window is design to help users easily load and display shockwave image sequences, as shown in **Figures 8-10**. Different edge detection methods can be chosen to detect the shockwave. Edge detection thresholds can be specified by users or automatically chosen by the program. Users can select to display random frame or play the image sequence continuously. In either case, the edge detection results will be displayed at the same time.

Two sequences are tested using the software developed. Based on the edge detected using edge detection methods, shockwave thickness is measured, results can be seen in **Table 1**. From the results, we can see that, even though they are different from each other, the edge detection methods perform exactly the same. For sequence #1, all methods measure the shockwave thickness as 10 pixels; while for sequence #2, all methods agree that the shockwave thickness is 9 pixels. Users can select edge detection method according to their applications or the characteristics of the image sequence. For example, if the sequence has a higher level of noise, then “Canny” or “LoG” can be used due to their noise resistance property. However, these two methods have higher computational complexities. For quick detection applications, the other three methods may be more suitable.


windtunnel

Original Images: avi022

Processed Images

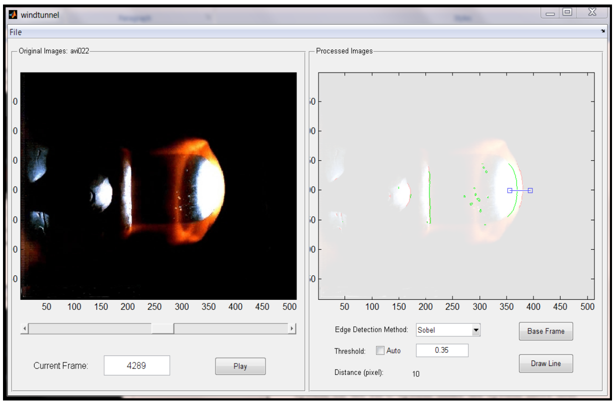
Edge Detection Method: Roberts

Threshold: □ Auto 0.1

Distance (pixel): 10

Current Frame: 4289

Figure 8. Shockwave detection with “Roberts”.


windtunnel

Original Images: avi022

Processed Images

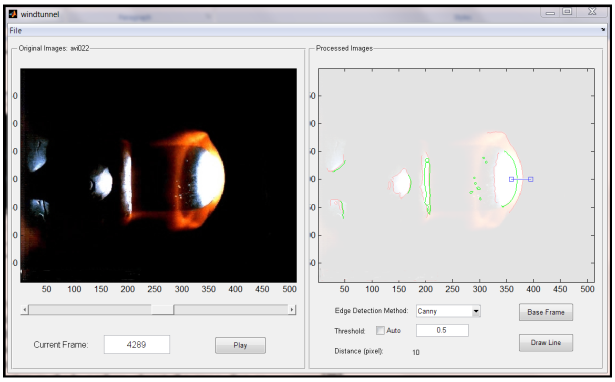
Edge Detection Method: Sobel

Threshold: □ Auto 0.35

Distance (pixel): 10

Current Frame: 4289

Figure 9. Shockwave detection with “Sobel”.


windtunnel

Original Images: avi022

Processed Images

Edge Detection Method: Canny

Threshold: □ Auto 0.5

Distance (pixel): 10

Current Frame: 4289

Figure 10. Shockwave detection with “Canny”.

Table 1. Shockwave thickness detection results.

Edge Detection Method	Shockwave Thickness (pixels)	
	Sequence #1	*Sequence #2*
Roberts	10	9
Prewitt	10	9
Sobel	10	9
Canny	10	9
Laplacian of Gaussian	10	9

a. All edge detection methods choose thresholds automatically.

5. Conclusions, Challenges, and Future Works

A shockwave detection system is built. The system utilizes image edge detection algorithms to easily and accurately detect shockwave. Based on the experimental results, the system meets the design requirements. However, there exists some challenges. The biggest challenge right now is how to deal with the large volume of data captured by high speed camera as well as the high computational complex image processing algorithms. To load the whole sequence in computer, 8 GB main memory is required. To accomplish real time processing, high-performance computer system is desired. The future works include testing this software system on the HPC system make the system more user friendly.

6. Acknowledgements

This work was partially supported by the NSF SEED award.

REFERENCES

[1] L. G. Roberts, “Machine Perception of Three-Dimensional Solids,” *Optical and Electro-Optical Information Processing*, J. T. Tippett, *et al.*, Eds., May 1965. http://www.packet.cc/files/mach-per-3D-solids.html#*

[2] J. M. S. Prewitt, “Object enhancement and extraction,” *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds., New York: Academic Press, 1970, pp. 75-149.

[3] I. E. Sobel, “Camera Models and Machine Perception,” Stanford Doctoral Dissertation, 1970.

[4] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679-698. doi:10.1109/TPAMI.1986.4767851

[5] D. C. Marr and E. Hildreth, “Theory of Edge Detection,” in *Proceedings of the Royal Society of London*, Vol. 207, No. 1167, Feb. 1980, pp. 187-217. doi:10.1098/rspb.1980.0020