CMP_SC 8690 - Computer Vision
HW1B: Color Edge Detection
Elizabeth Nash
2/13/2025

Abstract

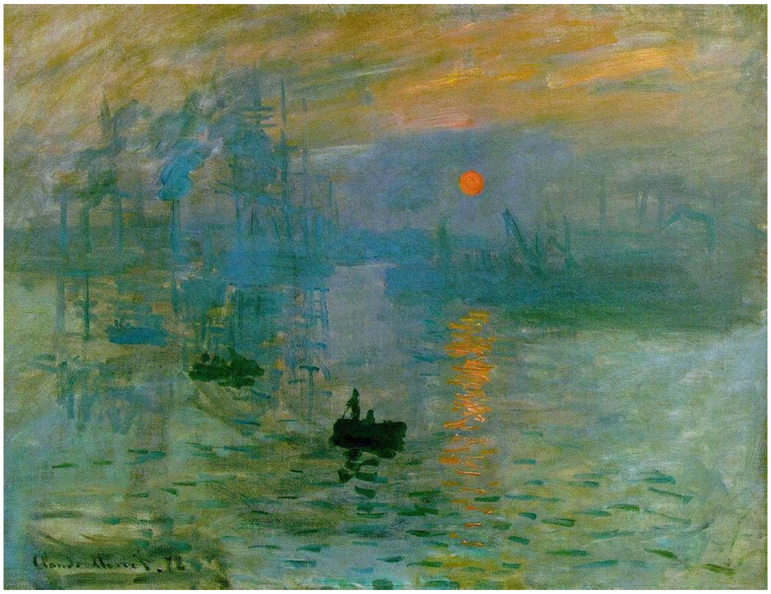
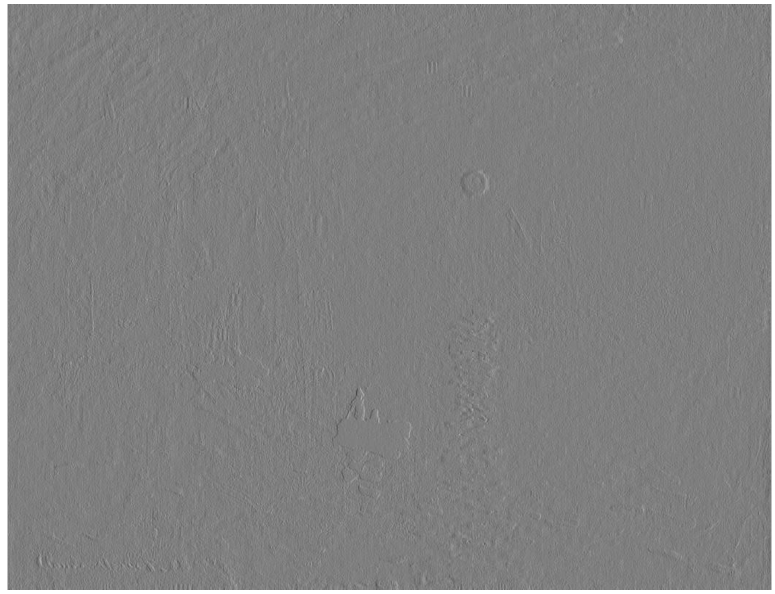
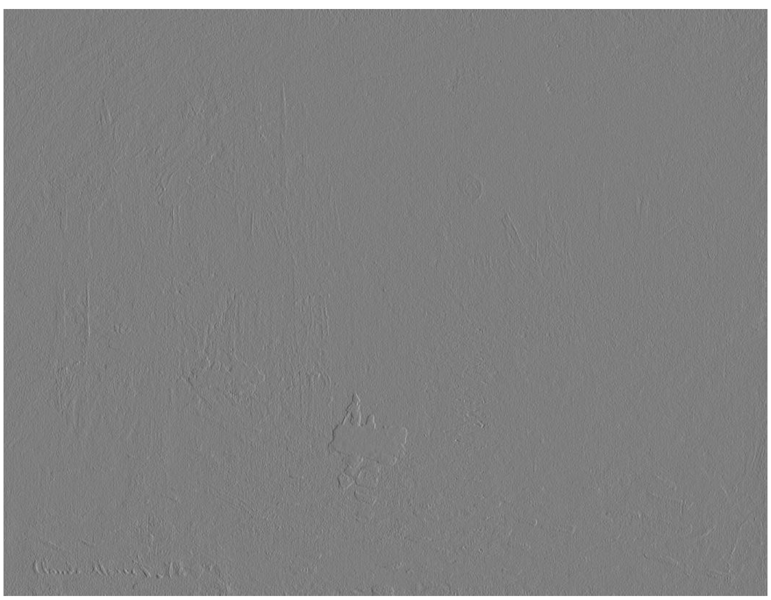
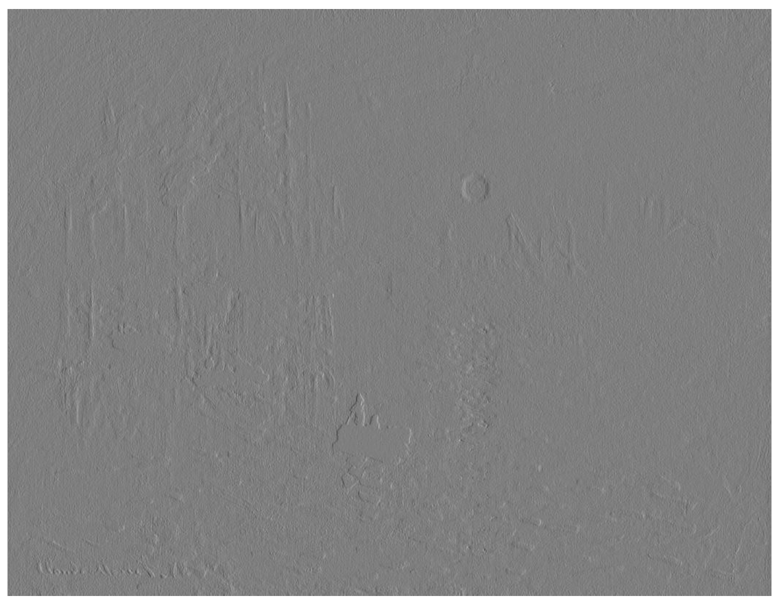
Details that are visually obvious may not be apparent in the numeric representation of images. In color images, edges may be the result of a combination of differences across multiple channels. Traditional single-channel edge detection techniques are unable to manage these situations. More advanced pipelines incorporate information across multiple channels to extend edge detection to color images.

Introduction

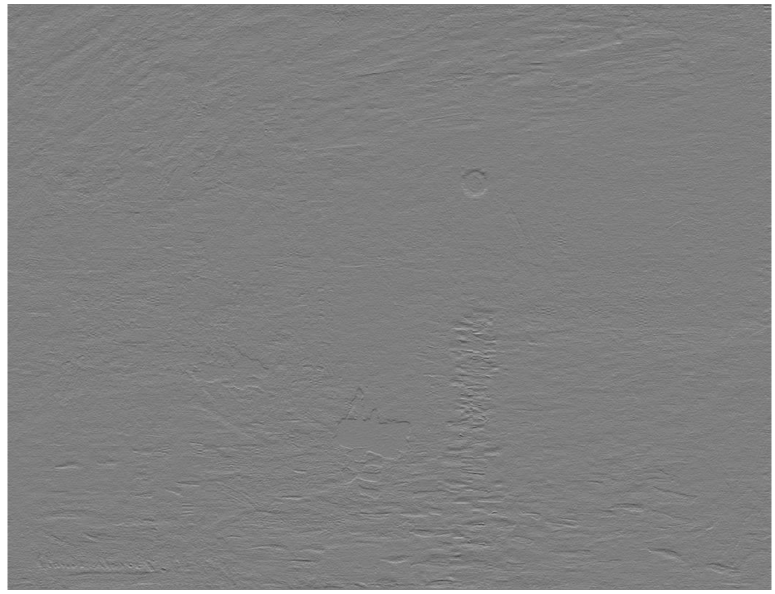
To perform edge detection on a color image, I first apply a Sobel filter to each channel in both the x- and y-directions to approximate the image gradient. Points where the color's intensity suddenly changes are highlighted, reflecting the edges for that channel. These gradients are combined to calculate the 2D Color Structure Tensor, which enables multi-channel edge detection.

All intensities for the below images were remapped to the display range of an 8-bit image: [0, 255]. Since the tensor element results only occupy a small portion of their possible value range, I also applied a Linear Contrast Stretch.

Results and Discussion

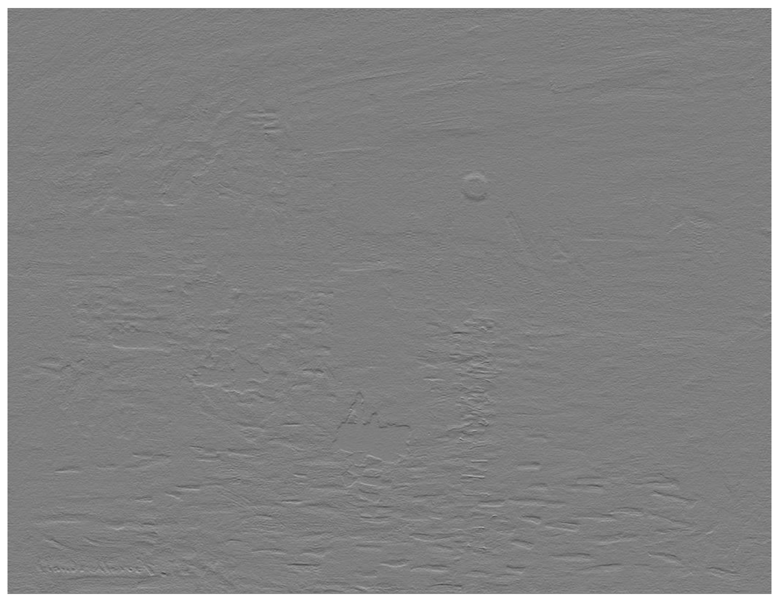
 RGB Input	 Blue Channel x-Gradient
 Green Channel x-Gradient	 Red Channel x-Gradient

These gradients were produced through convolution with a 1x3 Sobel filter, [-1, 0, 1] in the x-direction and rotated 90° for y. Larger sizes were avoided as they introduce a level of smoothing, which would alter the results of the 2D Color Structure Tensor.
These filters best detect edges perpendicular to their direction, hence the vertical lines in the x-gradient and horizontal lines in the y-gradient. Diagonal edges are lightly detected in both directions, such as the brush strokes in the top-left corner. Each channel's gradient reveals different details in the image. The blue and red channels, for example, clearly show the sun, whereas its green intensity only marginally differs from the sky.
The process to remap the gradients to the visual range causes the average gray tone, as a zero-gradient value is translated to 127.

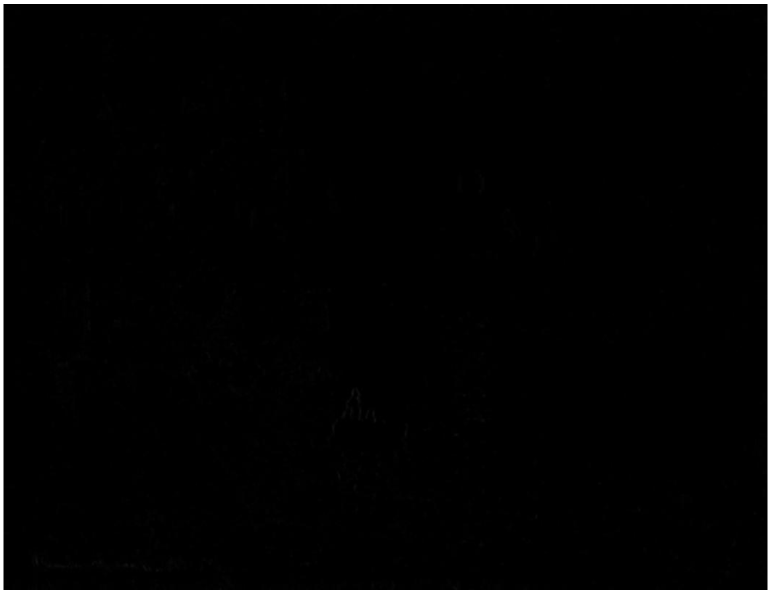
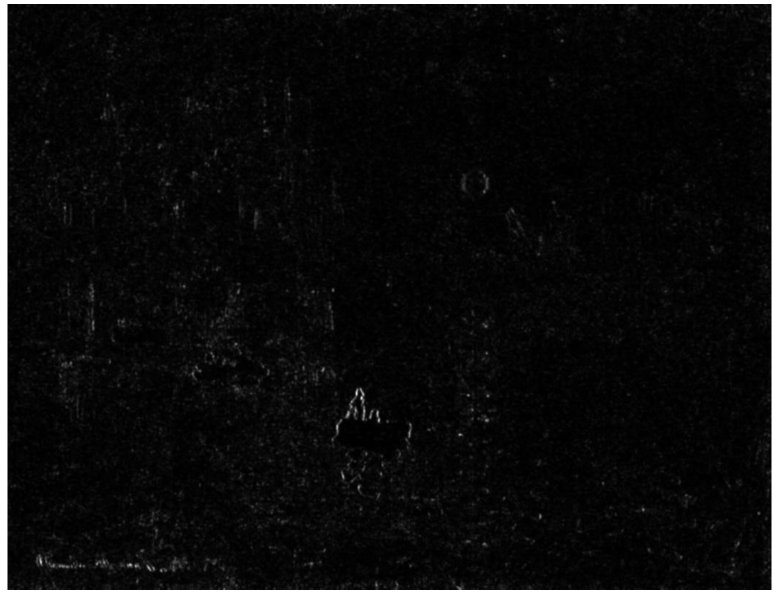
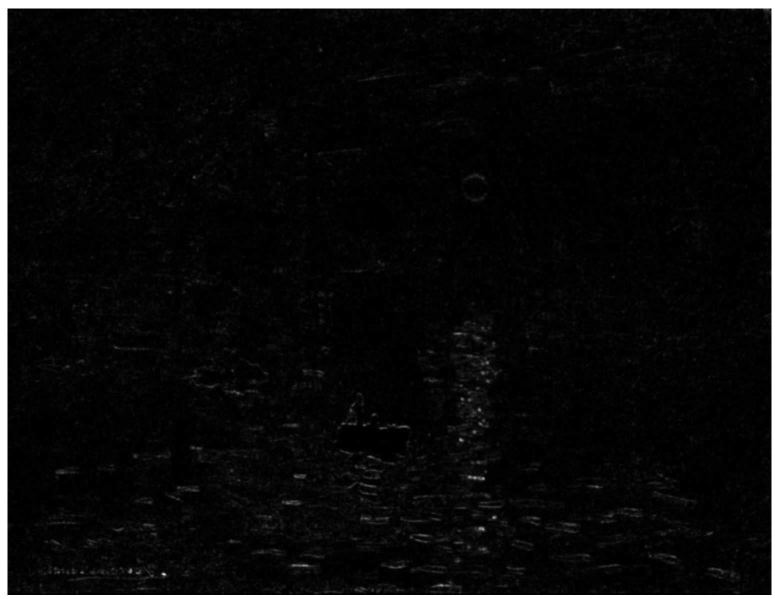


Blue Channel y-Gradient

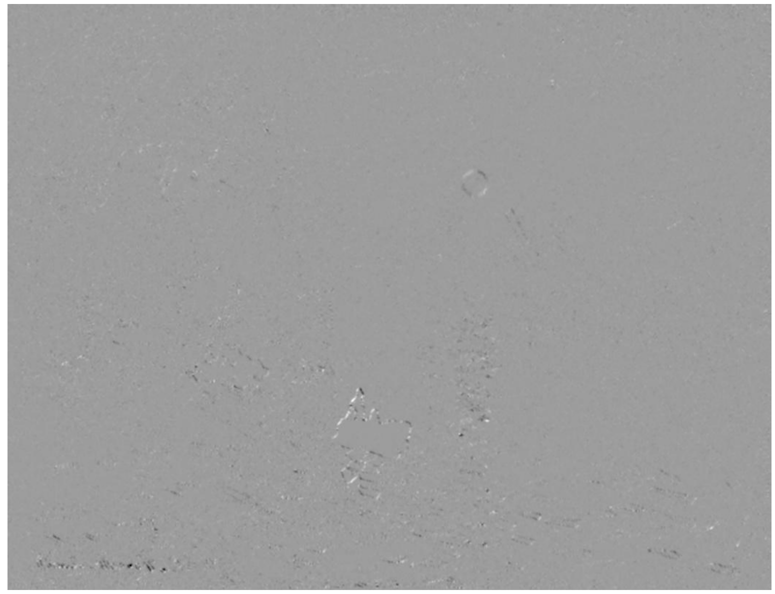
Green Channel y-Gradient



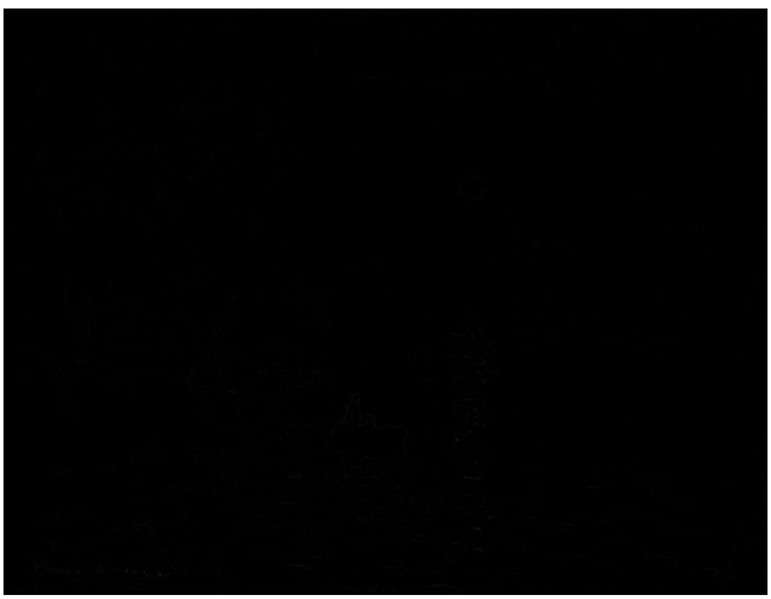
Red Channel y-Gradient

 Top-Left Tensor Element (Sigma 1)	 Top-Left Tensor Element LCS (Sigma 1)
 Bottom-Right Tensor Element (Sigma 1)	 Bottom-Right Tensor Element LCS (Sigma 1)

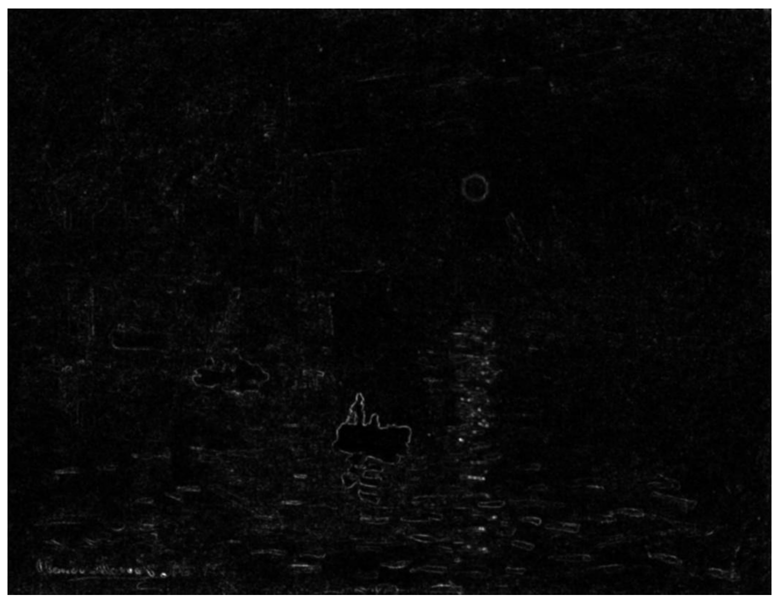
Top-Right/Bottom Left Tensor Elements
(Sigma 1)



Top-Right/Bottom Left Tensor Elements LCS
(Sigma 1)



Tensor Trace
(Sigma 1)

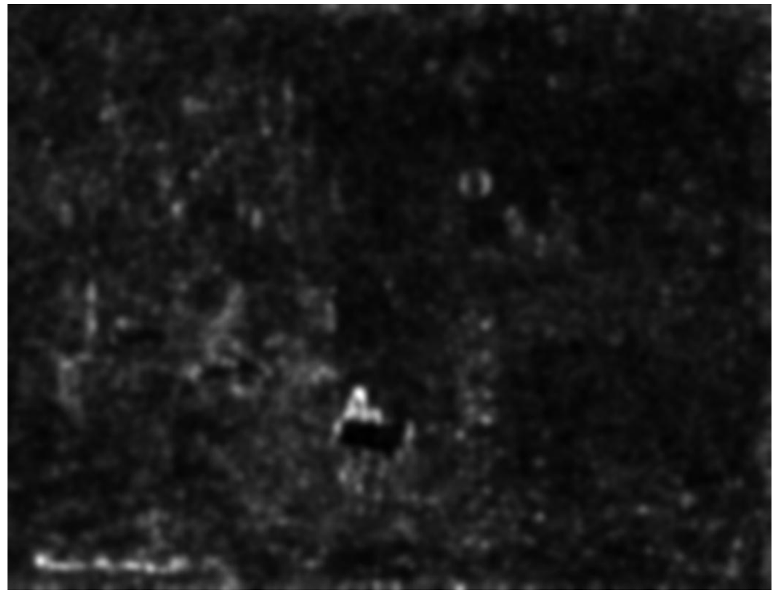


Tensor Trace LCS
(Sigma 1)

The 2D Color Structure Tensor’s four elements are different combinations of the x- and y-gradients. For each color channel, the gradients are either squared or multiplied together then convolved with a Gaussian filter. Every channel’s result is summed, resulting in one of the tensor elements. Due to the associative property, the top-right and bottom-left elements are equivalent.
Adding the top-left and bottom-right elements together produces the Tensor trace. These elements are preferred since the squaring process preserves only the magnitude of the edge, discarding the direction of change. Otherwise, opposing changes in different channels may add to zero. This difference is why the average tone of the top-right and bottom-left elements is gray (since 0 maps to 127) whereas the average tone is black for all others (since 0 maps to 0).

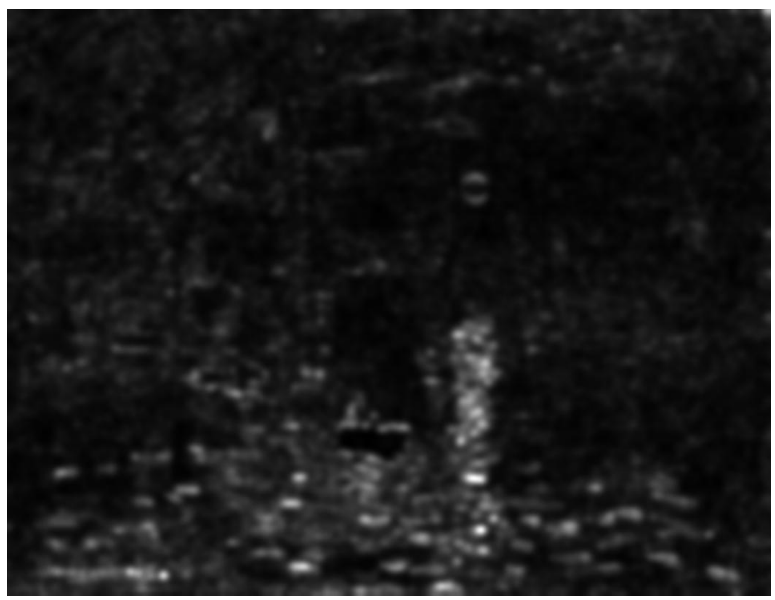
In order to preserve as much information as possible throughout the calculation of the Tensor, no data type conversions are performed beyond an initial translation into 64-bit signed numbers. As such, the possible value range can be as large as 6 million wide. The actual Tensor results occupy so little of this range that almost no detail is discernible with a direct remapping to displayable values. Therefore, a Linear Contrast Stretch was performed as well.

Top-Left Tensor Element
(Sigma 5)



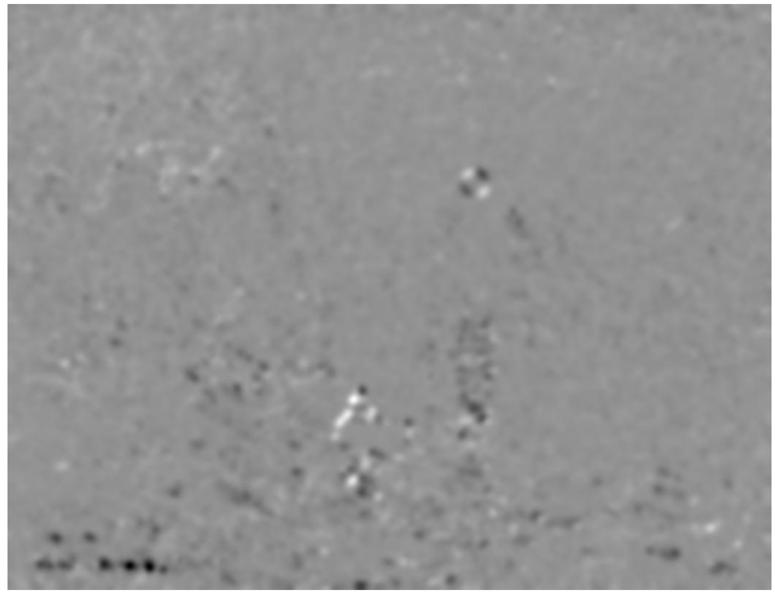
Top-Left Tensor Element LCS
(Sigma 5)

Bottom-Right Tensor Element
(Sigma 5)



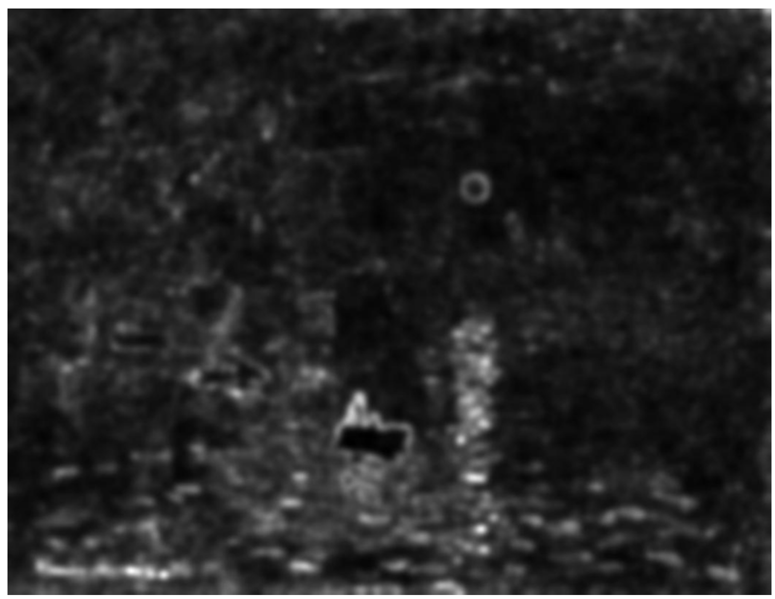
Bottom-Right Tensor Element LCS
(Sigma 5)

Top-Right/Bottom-Left Tensor Elements
(Sigma 5)



Top-Right/Bottom-Left Tensor Elements LCS
(Sigma 5)

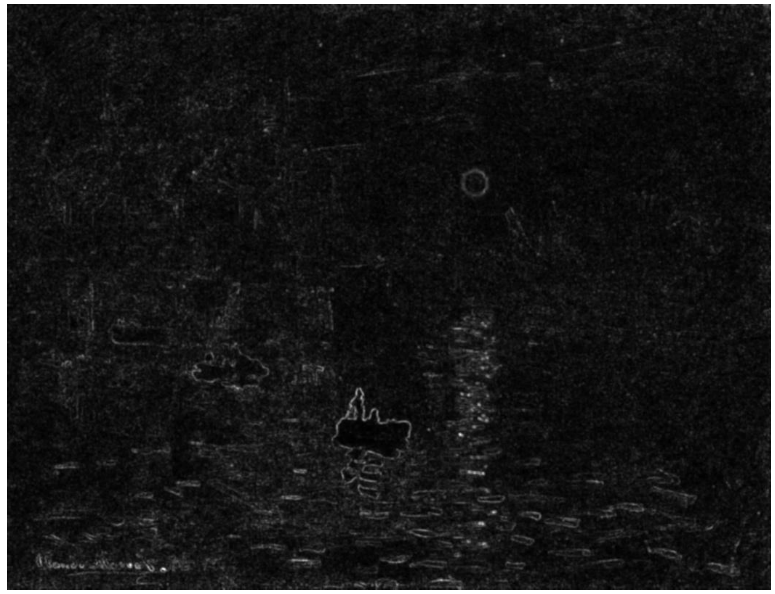
Tensor Trace
(Sigma 5)



Tensor Trace LCS
(Sigma 5)

Contrary to their typical usage, higher sigma values capture more of the image's details. This is because Gaussian filters have a purely amplifying effect. The same lines register in the Tensor with a sigma of one, but not at a high enough value to be visible. More of the images' individual edges can be viewed in the Tensor with a sigma of 5. The side effect of the higher Gaussian is that it applies greater smoothing, meaning overall structures are harder to discern.

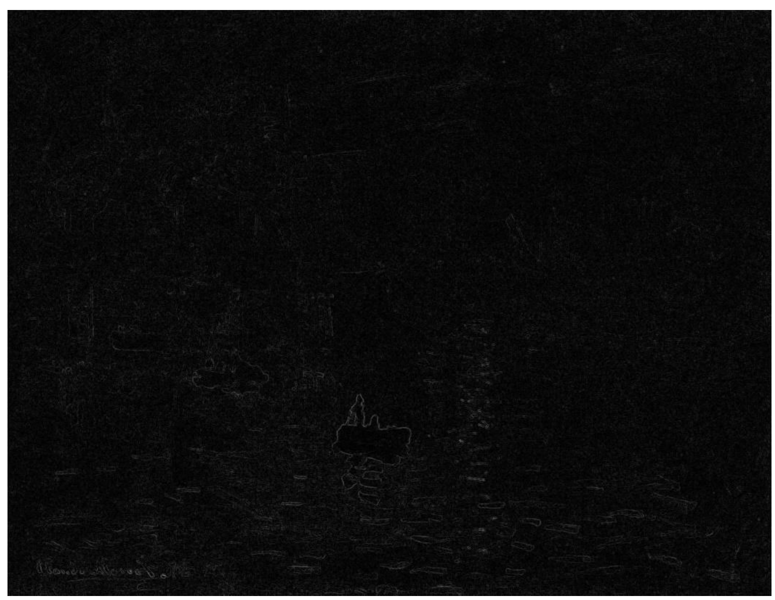
Adjusted Tensor Trace
(Sigma 1)
(Adjustment Factor .7)



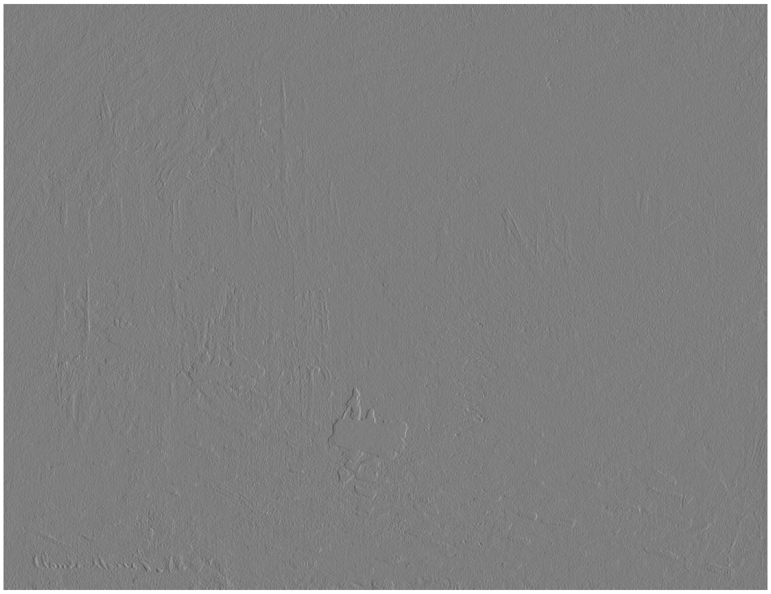
Adjusted Tensor Trace LCS
(Sigma 1)
(Adjustment Factor .7)

The unequal amplification of the Gaussian filter is not the only way to increase the visibility of details. Decreasing the highest end of the value spectrum allows the Linear Contrast Stretch to display more of the middle range, instead of collapsing it to the same value. As such, I adjusted the sigma 1 Tensor by raising its values to a power of .7. More of the image's edges are revealed without the smoothing of a higher sigma.

Grayscale Input



Grayscale Gradient Magnitude



Grayscale x-Gradient

Grayscale y-Gradient

Finally, gradient processing was performed on a grayscale version of the image. The gradient magnitude functions similarly to the Tensor trace, by combining the x- and y-gradients. Many of the same edges are captured, except those distinguished solely by single-color shifts. This is best illustrated by the sun, which does not appear in either gradient.