Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Image Processing in Repairing the Red Zone of Vehicle Barriers in Binjai City with Edge Detection Algorithm

Sidi Uzayva Resyad[1]*, Hotler Manurung[2], I Gusti Prahmana[3]

[1,2,3] STMIK Kaputama,Indonesia
jivaresyad2000@gmail.com[1*], manurunghotler0@gmail.com[2],igustiprahmana4@gmail.com[3]

Abstract

Red zones on road markings are an important element in the traffic system that serves as a barrier or prohibition on stopping, parking, or crossing certain areas. In Binjai City, red zones are commonly found at intersections, near zebra crosses, or busy areas such as markets and schools. However, in its implementation in the field, the effectiveness of red zones is often not optimal due to various obstacles. In addition, "manual surveillance of red zone conditions" requires large human resources and has not been able to reach all vulnerable points effectively. Regular checks and maintenance efforts are often hampered by time and budget constraints. As a result, some red zone points are damaged or lost unnoticed for a long time.
This study aims to design and test image processing methods with edge detection algorithms in detecting and improving the appearance of traffic red zones in Binjai City. It is hoped that this solution can increase the effectiveness of traffic supervision and support efforts to control highways in a more modern and efficient manner. The result of the calculation above is a binner image with the number 0 being the color that shows black and the number 1 is the color that shows white. Showing the image is the result of a black and white image process. So from the calculation above, there is a Sobel algorithm that calculates the final value of the higher calculation is the Sobel algorithm with the level of fineness and clarity in the image.

Keywords: *Citra, Zona Merah, Edge Detection*

1. Introduction

Red zones on road markings are an important element in the traffic system that serves as a barrier or prohibition on stopping, parking, or crossing certain areas. In Binjai City, red zones are commonly found at intersections, near zebra crosses, or busy areas such as markets and schools. However, in its implementation in the field, the effectiveness of red zones is often not optimal due to various obstaclesOne of the main problems is the "fading of traffic red zone marking lines", both due to weather factors, paint age, and heavy vehicle traffic. Many red zones are no longer clearly visible to motorists, so they are often ignored or violated. Generally, edge detection is the initial process of image segmentation leading to the identification of objects contained in the image. Image segmentation is part of the image processing process which is a way to divide an image into regional groups that aim to isolate an object in the image. Objects that have been segmented can be carried out an image feature extraction process, which is a step that aims to distinguish between one object and another. This study will use the Sobel method to detect the edges of moving images (real-time images) in the form of images. [6]

The Sobel algorithm is a development of Robert's method which uses an HPF (High Pass Filter) filter that is given a zero on the buffer. This method uses the principle that can generate HPF, which is the principle of Laplacian and Gaussian functions. The advantage of the Sobel algorithm is the ability to reduce the amount of noise before performing edge detection calculations [1].Although the Sobel Operator has some limitations, such as sensitivity to noise and is less than optimal for detecting smooth edges, the advantages in its simplicity and effectiveness make it still relevant and widely used in practice.[7]

2. Theoretical Foundations

2.1. Image Processing

Image processing is one of the fields of science and technology that can be used to improve image quality and process image recognition to enrich the information needed from the processed image. Even though an image is rich in information, often an image will experience a decrease in quality intensity, such as having noise, colors that are too contrasting, of course an image like this will be difficult to present so as to reduce the information that will be obtained.[5]

2.2. Format Citra

Digital images are not only stored on storage disks, but also begin to spread for various purposes. In the process of sending data, the amount of data must of course be considered, so that data compression is required which leads to the emergence of various digital image formats .[4]

2.3. Traffic Zones

Models need to show that they provide an accurate picture of current travel before they should be used to forecast future travel. Better data, better representation of bicycles and pedestrian trips, better auto-occupancy models, better time of day factors, using more destination trips, better representation of access, putting costs into the distribution of trips, adding feedback land use, adding intersection delays – some of the key points that should be considered and included in traditional transportation modeling systems to make it far more comfortable and realistic. The determination of traffic zone boundaries requires consideration of several criteria. The commonly used criteria consist of: spatial proximity, i.e. the units that make up the traffic zone must be close to each other, homogeneity, is the type of land use and socio-economic characteristics that are similar in one zone, compactness, meaning that the shape of the traffic zone must be slender and spatially dense, there must be no zones that are in other zones, and administrative boundaries, Main roads, railroads, rivers, and other physical boundaries are used as zone boundaries where possible. The criteria of homogeneity and proximity are sometimes mutually exclusive and difficult to meet at the same time. Therefore, a systematic framework must ensure that the required criteria are adopted in a harmonious manner to achieve generation optimality.[8]

Fig. 1 : School Red Zone

2.4. Edge Detection

Edge detection is a sudden (large) change in the intensity value of the gray degree in a short distance. Edge detection is run to detect the Edge that delimits two homogeneous image regions that have different brightness levels. Image edge detection is also a process to generate edges from image objects with the primary purpose of finding and identifying sharp discontinuities of an image. Edge detection is also used mostly to get information from the frame as a first step for feature extraction and object segmentation. Edge detection can be divided into two groups. The first group is called first-order edge detection, this group works by using first-order derivatives or differentials i.e., Roberts, Prewitt, and Sobel operators. The second group is called second-order edge detection, which uses second-order derivatives. An example that falls into this group is the Laplacian of Gaussian (LoG) (Khairunnisa et.al, 2024).

The fourth study entitled "Edge Detection on Ultrasound Images of COVID-19 Disease Using the Sobel Method" The purpose of this study is to make a simulation on the COVID-19 X-ray image and determine the area of infection of the virus using the sobel method. The reason for using the sobel method in this study is because it has the advantage of completing calculations easily and producing images that are easy to read compared to several other methods. The benefits obtained from this research are so that the results of the images are clear, so that they can read the COVID-19 X-ray images carefully and be proven to be true, then they can understand and add to the science, especially edge detection using the sobel method.[3]

An image or image in Latin imago is a representation, resemblance, or imitation of an object or thing. Images can be grouped into visible images and invisible images. Examples of images appear in everyday life: photographs, drawings, and paintings, while invisible images are e.g.: image data in a file (digital image), and the image is represented as a mathematical function. Among these types of images, only digital images can be processed using computers. Other types of images, if they are to be processed by computer, must first be converted into digital images, for example, photos are scanned with a scanner, the body heat distribution of photos is captured with an infrared camera and converted into numerical information, density information and the composition of the inside of the human body are captured with the help of x-ray planes and radiation detection systems into digital information. The activity to convert non-digital physical image information into digital is called imaging. Digital imagery can be defined as a function of two variables, f(x,y), where x and y are the spatial coordinates and the value f(x,y) is the intensity of the image at those coordinates, The basic technology for creating and displaying colors in digital images is based on research that a color is a combination of three basic colors, namely red, green, and blue (Red, Green, Blue - RGB).[2]

3. Troubleshooting Methods

This method contains a description of the stages of the study implementation and a description of the analysis method used.

3.1. Algoritma Sobel

Sobel's Edge Detection, like the evolution of Prewitt's operator, is an improvement over the previous edge detection method (Robert's method) which used HPF (high pass filter) with zero buffer allocation. This algorithm contains a programming algorithm that acts as an image filter. This filter finds all the existing edges. This filter uses an operator called Sobel. The kernel size that the Sobel Method uses is 3x3 pixels to calculate the gradient, so the approximate slope is in the center of the window. Such a matrix is used to get the central pixel, thus becoming the center of the matrix. Using this matrix is like using a raster, which is filling the matrix with the desired pixels around it (the middle pixel). Suppose the pixel array is around pixels (x,y).

The steps for calculating *the Sobel* Edge Detection Algorithm are below:

1. When the image is smoothed, the derivatives Ix and Iy wrt x and y are calculated. This can be implemented by combining I with *the Sobel* kernels Kx and Ky, with the equation

$$K_x = \begin{bmatrix} -1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad K_y = \begin{bmatrix} -1 & 2 & 1 \\ -2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \tag{1}$$

Information:
Kx: Horizontal
Ky: Vertical

2. Then, the magnitude G and the slope θ of the gradient are calculated by

$$G = \sqrt{Gx^2 + Gy^2} \tag{2}$$

3. Determine the Thresholding value using the equation

$$g(x, y) = \begin{cases} 1, \text{If } f(x, y) > 128 \\ 0, \text{If } f(x, y) < 128 \end{cases} \tag{3}$$

Information:
By checking the number if the number value is >128 then the binary value given is 1, if the number value is <128 then the binary value given is 0.[5]

3.2. Research Supporting Data

The red zone has two different types, namely the traffic red zone and the Zoss (School Safe Zone). One of the differences is the symbol and size of the red zone. The traffic red zone has a zone size that is not large enough because it only covers the border between the train and car zones, while the school safety zone (zoos) has a large enough area because the area indicates a special area around the school that is marked or made special arrangements to improve the safety of students coming and going from school.

Fig.2 : Traffic Red Zone

3.2.1. Algorithm Analysis

Based on the observations that have been made, the detection of the edge to analyze the damage to the red zone through its color. Judging from the color of the red zone, there are more similarities that will be explained in the picture in this article.

Fig. 3: Zona Merah

a. Konvolosi Matrix

Matrix convolution is the process of converting an image into a matrix number to get the image into a number. Makes it easier to calculate blocks with pixel size in the *Sobel Algorithm.* The following is an image that has a black image putih pada zona merah Untuk Performing the edge calculation process, the image is converted to black and white. The black and white image of the red zone can be seen as follows:

Fig. 4: Black and White Red Zone

The division of the image into a size of 10 x 10 pixels to adjust to *Sobel*, then the size of the processed image will be broken down into blocks with a size of 10 x 10 pixels per block as below:

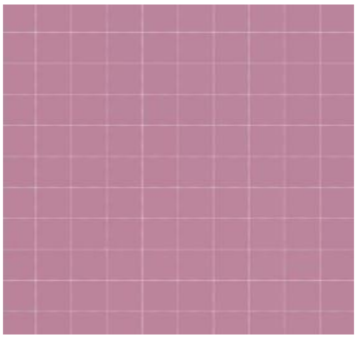


Fig. 5: Pixel Zona Merah

b. Sobel Edge Detection Method Calculation

Determine *the Sobel Edge Detection operators Kx* and *Ky* use the matrix of *the Sobel Edge Detection operators Kx and Ky* to perform edge detection.

$$K_x = \begin{bmatrix} -1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \qquad K_y = \begin{bmatrix} 1 & 2 & 1 \\ -2 & 0 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

These pixel values will be processed using *the Sobel Edge Detection* method to detect the edges in the image. The pixel values of the image above are processed according to the provisions of the *Sobel Edge Detection method.* Convolution *of grayscale* imagery with horizontal (Kx) *Sobel Edge Detection* kernel and vertical *Sobel Edge Detection* (Ky) kernel.

Determine the gradient value of gray imagery that has been operated with the mask matrix via *Sobel Edge Detection*

206	210	210	203	197	198	202	205	214	217
199	204	205	200	196	198	203	205	214	217
189	196	200	197	196	200	206	209	215	218
178	186	194	195	197	203	210	213	217	220
165	176	186	191	196	204	212	215	220	222
155	168	181	189	196	205	214	217	223	225
155	169	184	194	202	212	221	224	225	227

Fig. 6: Image Pixel Number 10 x 10

Kx = (206) (-1) + (210) (0) + (210) (0) + (199) (-2) + (204) (1) + (205) (0) + (189) (-1) + (196) (0) + (200) (1) = -389

Ky = (206) (1) + (199) (-2) + (189) (-1) + (210) (2) + (204) (0) + (205) (-2) + (189) (1) + (196) (-2) + (200) (-1) = -774

K[f(x,y)]= $\sqrt{-389^2 + -774^2}$ = 866.38

The pixel values obtained from the pixel calculation operated by *the Sobel Edge detection* method at line point 1, column 1 are 866.38.

Kx = (205) (-1) + (214) (0) + (217) (0) + (205) (-2) + (214) (1) + (217) (0) + (209) (-1) + (215) (0) + (218) (1) = -392

Ky = (205) (1) + (205) (-2) + (209) (-1) + (214) (2) + (214) (0) + (215) (-2) + (217) (1) + (217) (-2) + (218) (-1) = -851

K[f(x,y) = $\sqrt{-392^2 + -851^2}$ = 936.46

The pixel value obtained from the pixel calculation operated by *the Sobel Edge detection* method at line point 1, column 8 is 936.46.

The final calculation is a pixel calculation process that continues until it is completed.

Determining the *Thresholding Value*

$$g(x, y) = \begin{matrix} 1, If\ f(x,y) \geq 128 \\ 0, If\ f(x,y) < 128 \end{matrix}$$

By checking the number if the number value is >128 then the binary value given is 1, if the number value is <128 then the binary value given is 0.

866.38	413.25	424.17	403.99	890.63	904.61	933.07	936.46	798.328	835.661
871.733	866.838	903.497	927.35	955.32	958.87	746.85	793.80	847.48	879.96
912.29	936.77	965.5	976.67	700.383	759.91	821.15	865.0	901.12	951.23
972.65	986.54	703.278	774.8.	836.71	885.29	925.53	960.82	998.20	1,001.80
752.58	817.13	884.84	937.25	977.32	1,012.65	1,036.73	1,029.24	786.65	840.78
904.18	957.86	1,003.17	1,052.06	1060.93	1,074	801.53	847.89	905.38	961.97
1,008.71	1,052.65	1,087.56	1,092.21	202	212	221	224	225	227
160	174	190	201	210	220	229	232	227	229
166	178	192	203	213	225	234	237	238	236
170	182	196	207	217	230	239	242	238	236

Fig. 7: Grayscale Sobel Edge Detection Image Results

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Fig. 8: Calculation Results of Binner Sobel Edge Detection

The result of the calculation above is a binner image with the number 0 being the color that shows black and the number 1 is the color that shows white. Showing the image is the result of a black and white image process. So from the calculation above, there is a *Sobel* algorithm that calculates the final value of the higher calculation is the *Sobel algorithm* with the level of fineness and clarity in the image.

4. Implementation of the Jupyter Notebook Program

The explanation of the interface design that has been made by the research with the *sobel edge detection* algorithm has results to see the results of the traffic red zone detection image program carried out by the research. The following is a discussion of interface design in the image system:

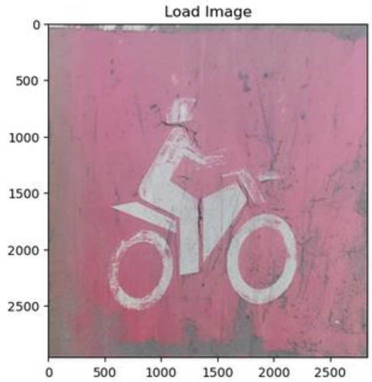
a. Display *Interface Library Jupyter Notebook*

```
[1]: # Import Library yang diperlukan
import cv2
import numpy as np
import imageio
import matplotlib.pyplot as plt
```

```
[2]: # Membaca Load Gambar
imgNormal = imageio.imread("zonamerah.jpg")
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_12680\1261280203.py:2: DeprecationWarning: Starting with ImageIO v3 the behavior of this function will switch to
that of iio.v3.imread. To keep the current behavior (and make this warning disappear) use `import imageio.v2 as imageio` or call `imageio.v2.imread` direct
ly.
  imgNormal = imageio.imread("zonamerah.jpg")
```

```
[3]: # Menampilkan gambar citra Load Gambar
plt.imshow(imgNormal)
plt.title("Load Image")
plt.show()
```




```
[63]: # Fungsi untuk menampilkan gambar
def show_image(title, image):
    cv2.imshow(title, image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

```
[64]: # Baca gambar (ganti 'your_image.jpg' dengan path gambar Anda)
image = cv2.imread('zonamerah.jpg', cv2.IMREAD_GRAYSCALE)
```

```
[65]: # Baca conpres data ke angka
image
```

```
[65]: array([[208, 211, 214, ..., 144, 143, 143],
       [205, 208, 211, ..., 144, 144, 143],
       [199, 203, 207, ..., 144, 144, 143],
       ...,
       [122, 123, 122, ..., 131, 133, 135],
       [122, 122, 122, ..., 133, 134, 135],
       [121, 121, 121, ..., 135, 134, 135]], dtype=uint8)
```

```
[66]: # Periksa apakah gambar berhasil dimuat
if image is None:
    print("Error: Gambar tidak ditemukan!")
    exit()
```

```
[67]: # 2. Sobel Edge Detection
# Hitung gradien dalam arah x dan y
sobel_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)  # Gradien arah x
sobel_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3)  # Gradien arah y
```

```
[68]: # Menggabungkan gradien x dan y untuk mendapatkan magnitudo
sobel_edges = np.sqrt(sobel_x**2 + sobel_y**2)
sobel_edges = np.uint8(sobel_edges / sobel_edges.max() * 255)  # Normalisasi ke 0-255
```

```
[72]: def show_image(title, image):
    plt.figure(figsize=(8, 6))  # ukuran dalam satuan inci (width, height)
    plt.imshow(image)
    plt.title("Original Image")
    plt.axis('off')
    plt.show()
```

```
[73]: def show_image(title, image, size=(8, 6)):
    plt.figure(figsize=size)  # ukuran dalam inci
    plt.imshow(image, cmap='gray')  # pakai cmap gray untuk hasil edge detection
    plt.title("Sobel Edge Detection")
    plt.axis('off')
    plt.show()
```

```
[74]: show_image("Original Image", image, size=(10, 10))
```

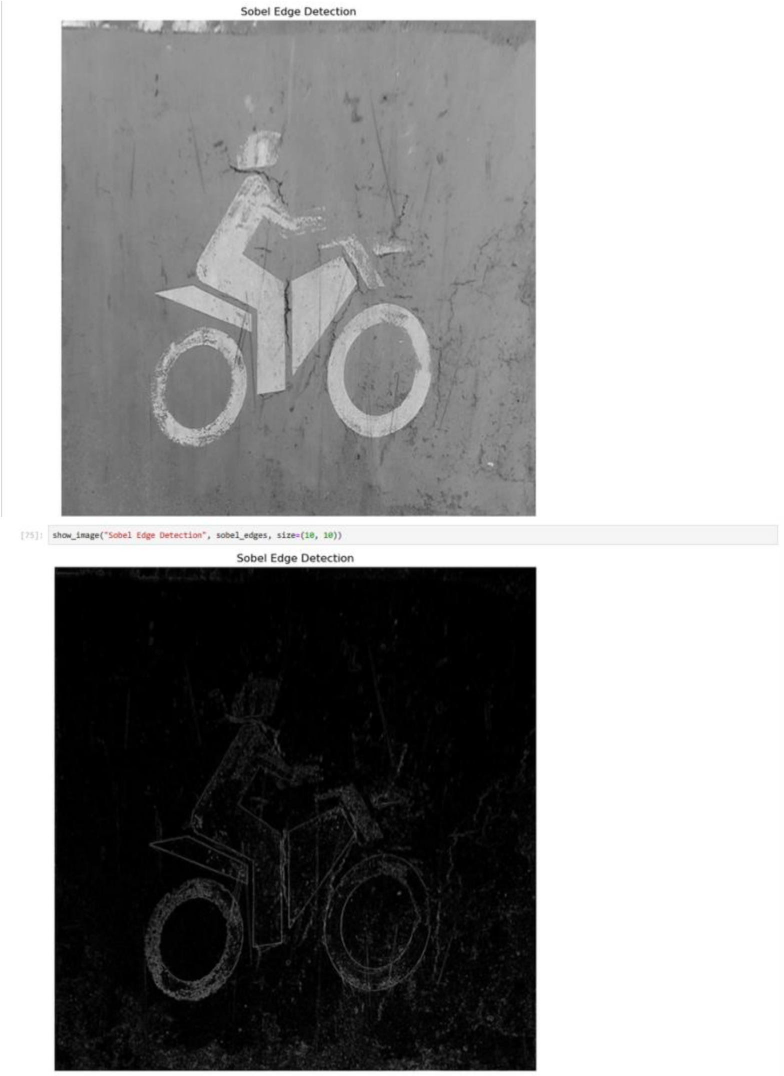


Fig. 9: Result Program Sobel Edge Detection

5. Conclusion

After explaining in the previous chapter. Then I will give some conclusions about the results of the program made. The following are the conclusions of the study related to the red zone image processing technique based on texture images with *the Sobel Edge Detection* Algorithm:

1. This research proves that *the Sobel Edge Detection* algorithm can be applied to effectively detect edges in traffic red zone images.
2. The results of image processing showed that *the Sobel method* was able to display the boundaries of the red zone clearly in the form of a black-and-white binary image.
3. The gradient calculation process using *the Sobel* operator produces consistent edge values that make it easier to identify damage or unevenness in the red zone.

References

[1] Fadhli, R., Fiky, Y. S., & Junartho, H. (2017). Penerapan Deteksi Gerak Pada Kamera Pengawas Menggunakan Application of Motion Detection on Security Camera Using. E-Proceeding of Engineering, 4(2), 1441–1448.

[2] Gazali, W., Soeparno, H., & Ohliati, J. (2014). Dalam Pengolahan Citra Digital. Jurnal Mat Stat, 12(2), 103–113.

[3] Ghozali, M., & Sumarti, H. (2020). Deteksi Tepi pada Citra Rontgen Penyakit COVID-19 Menggunakan Metode Sobel. Jurnal Imejing Diagnostik (JImeD), 6(2), 51–59. https://doi.org/10.31983/jimed.v6i2.5840

[4] Khairunnisa, F., Suryanata, M. G., & Prayudha, J. (2022). Pengolahan Citra Untuk Mendeteksi Tepi Citra Gigi Berlubang Menggunakan Metode Canny. x, 1–11.

[5] Maximillian, L., Riti, Y. F., Agung, M. A., & Palis, Y. J. (2023). Perbandingan Algoritma Sobel dan Canny untuk Deteksi Tepi Citra Daun Lidah Buaya. Komputa : Jurnal Ilmiah Komputer Dan Informatika, 12(2), 69–79. https://doi.org/10.34010/komputa.v12i2.10997

[6] Moelya, M. S., Ramadhan, P. S., & Suryanata, M. G. (2024). Perbandingan Metode Canny , Sobel , Dan Laplacian of Gaussian Dalam Mendeteksi Tepi Citra Objek Bergerak. 3, 450–460.

[7] Supiyandi Supiyandi, Trisatin Panggabean, Nuzul Ramadhan, Sri Ratna Dewi, & Salsabila Yusra. (2024). Deteksi Tepi Sederhana Pada Citra Menggunakan Operator Sobel. Repeater : Publikasi Teknik Informatika Dan Jaringan, 2(3), 43–56. https://doi.org/10.62951/repeater.v2i3.90

[8] Pribadi, O. S., Nurhadi, M., & Hermawan, B. A. (2023). Analisis Pembangunan Zona Lalu Lintas dengan Standard Deviation Control Kepadatan Area Terbangun Studi Kasus: Kabupaten Klungkung, Provinsi Bali. Jurnal Manajemen Transportasi & Logistik (JMTRANSLOG), 10(3), 293. https://doi.org/10.54324/j.mtl.v10i3.1250