

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Систем автоматического управления

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Техническое зрение»
Тема: «Границы на изображениях»

Студент гр. 1492

Старцев Н.А.

Преподаватель

Федоркова А.О.

Санкт-Петербург

2024

Цель работы: изучить методы дифференцирования изображений и выделения границ на изображениях

Задание 1

Протестировать работу оператора Собеля на изображениях клетки (вертикальные и горизонтальные линии) и косой штриховки.

Протестировать работу фильтра Лапласа на тех же изображениях

Задание 2

Протестировать работу детектора границ Кенни на изображениях с четкими и нечеткими границами.

Задание 3

Вам нужно смоделировать видеоряд с камеры, которая установлена на мобильном роботе. Робот снимает дорогу под некоторым углом, камера смотрит вперед. Задача робота - остановиться перед линией разметки.

Возьмите лист бумаги, маркер (или принтер) и смоделируйте такую разметку. Запишите видео, которое бы получал робот с такой камеры. На полученном видео выделите границы так, чтобы на следующих шагах можно было найти линию.

.

Примеры работы программы:
Задание 1.

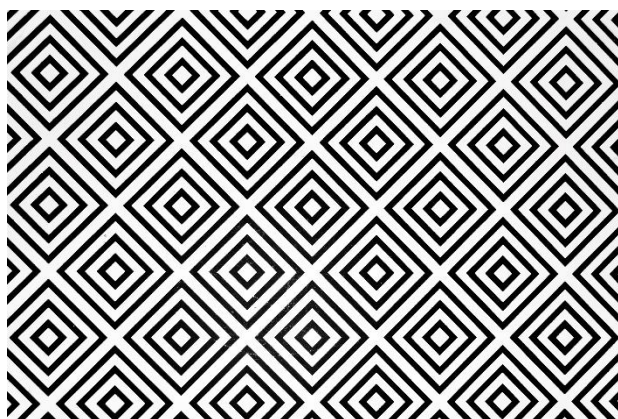
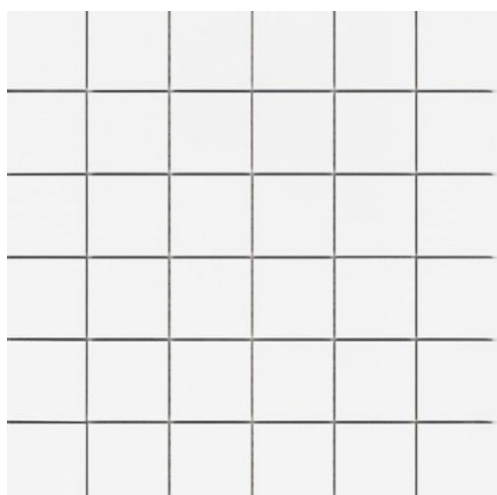


Рисунок 1 – исходные изображения

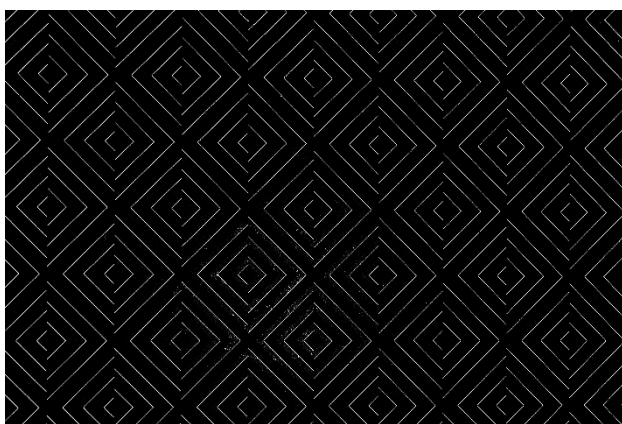
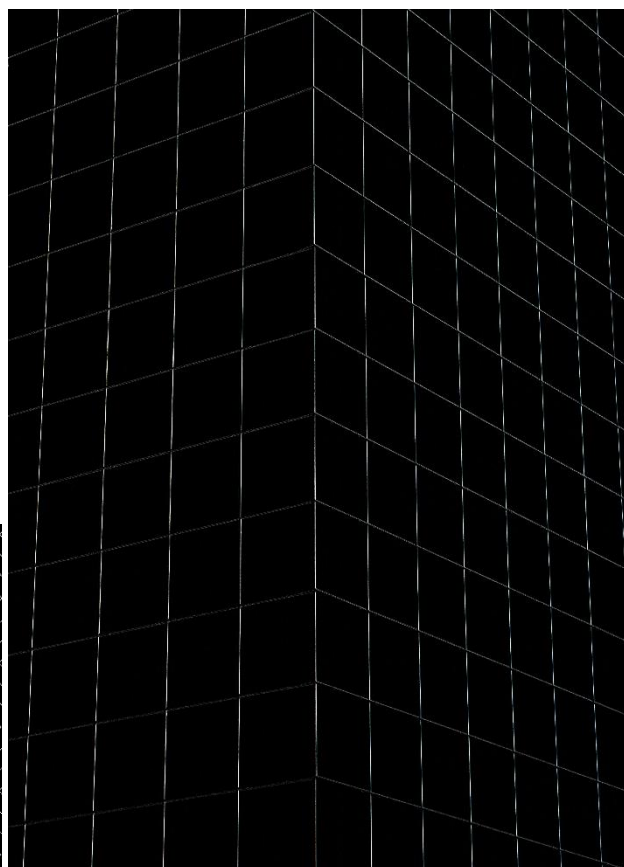
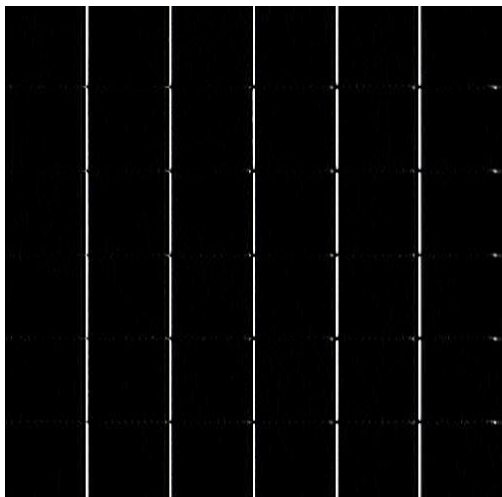


Рисунок 2 – пример работы оператора собеля, при $dx=1$ $dy = 0$

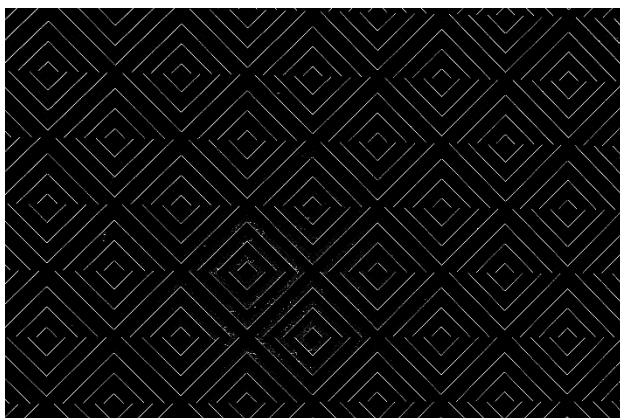
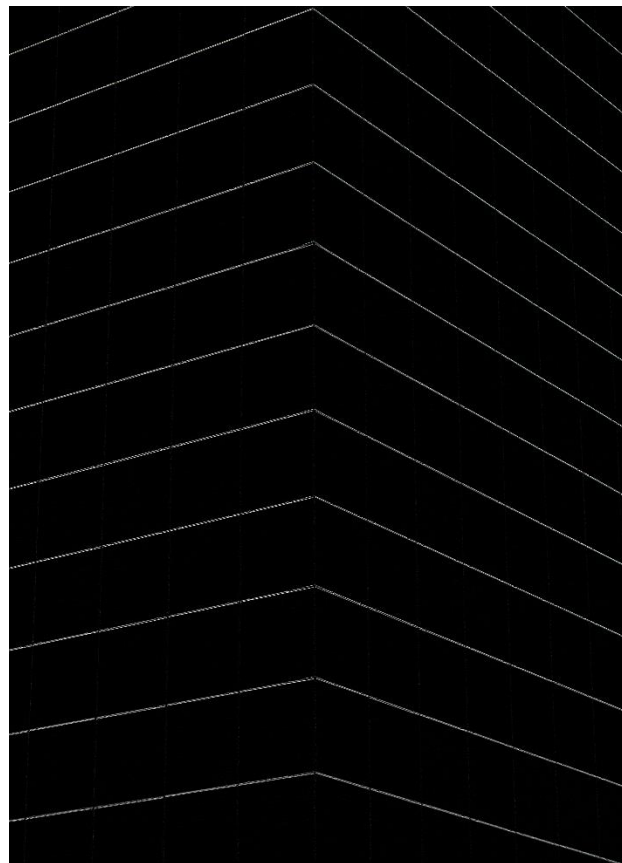
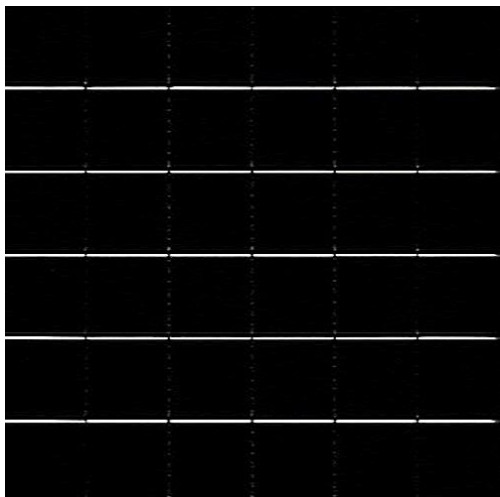


Рисунок 3 – пример работы оператора собеля, при $dx=0$ $dy = 1$

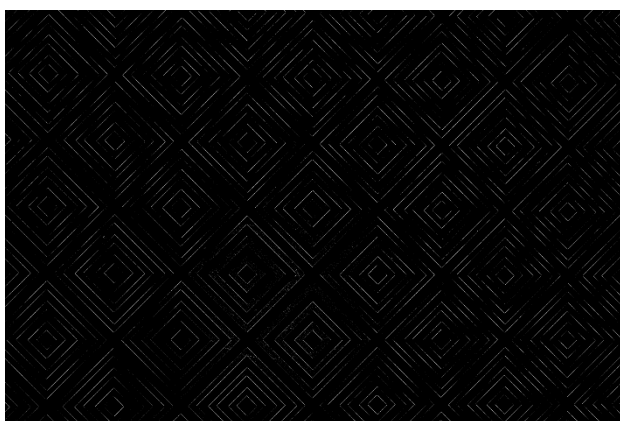
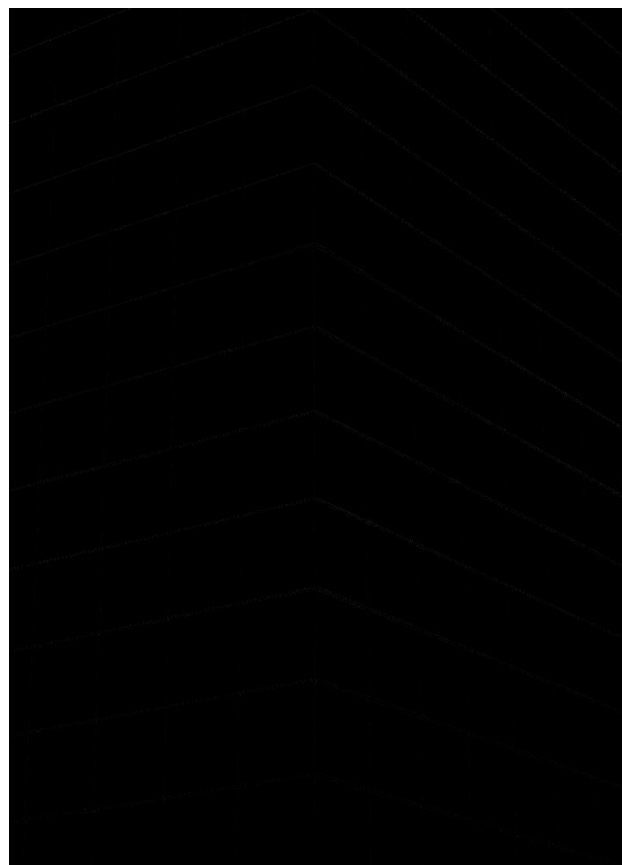
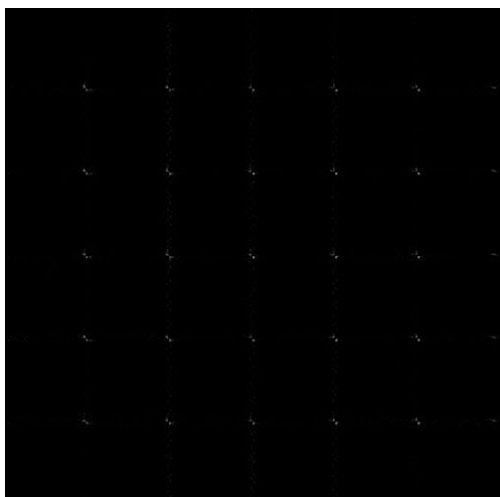


Рисунок 4 – пример работы оператора собеля, при $dx=1$ $dy = 1$

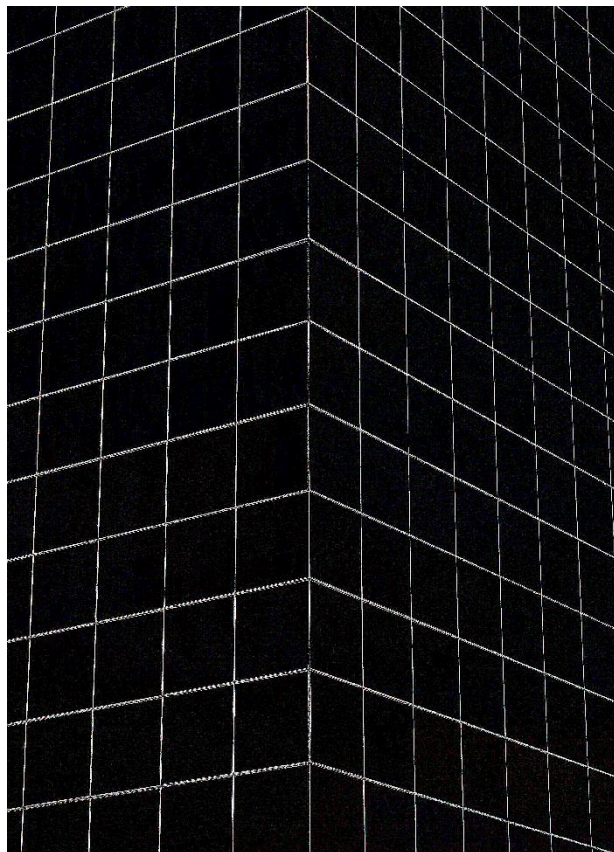
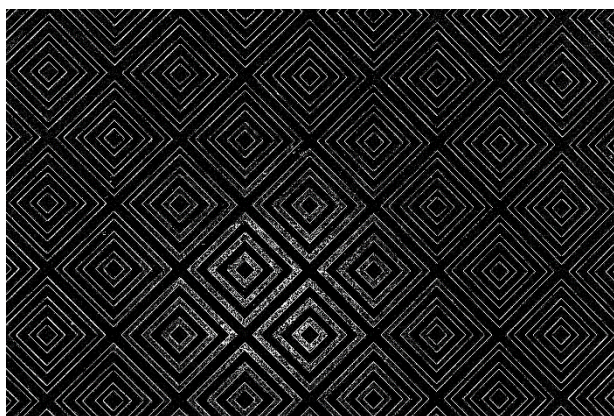
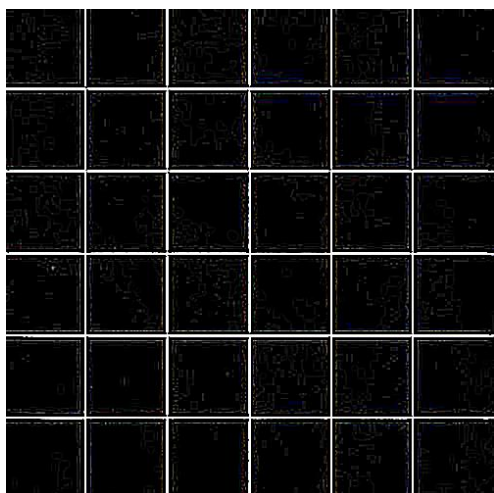


Рисунок 5 – пример работы оператора Лапласиана

Задание 2.

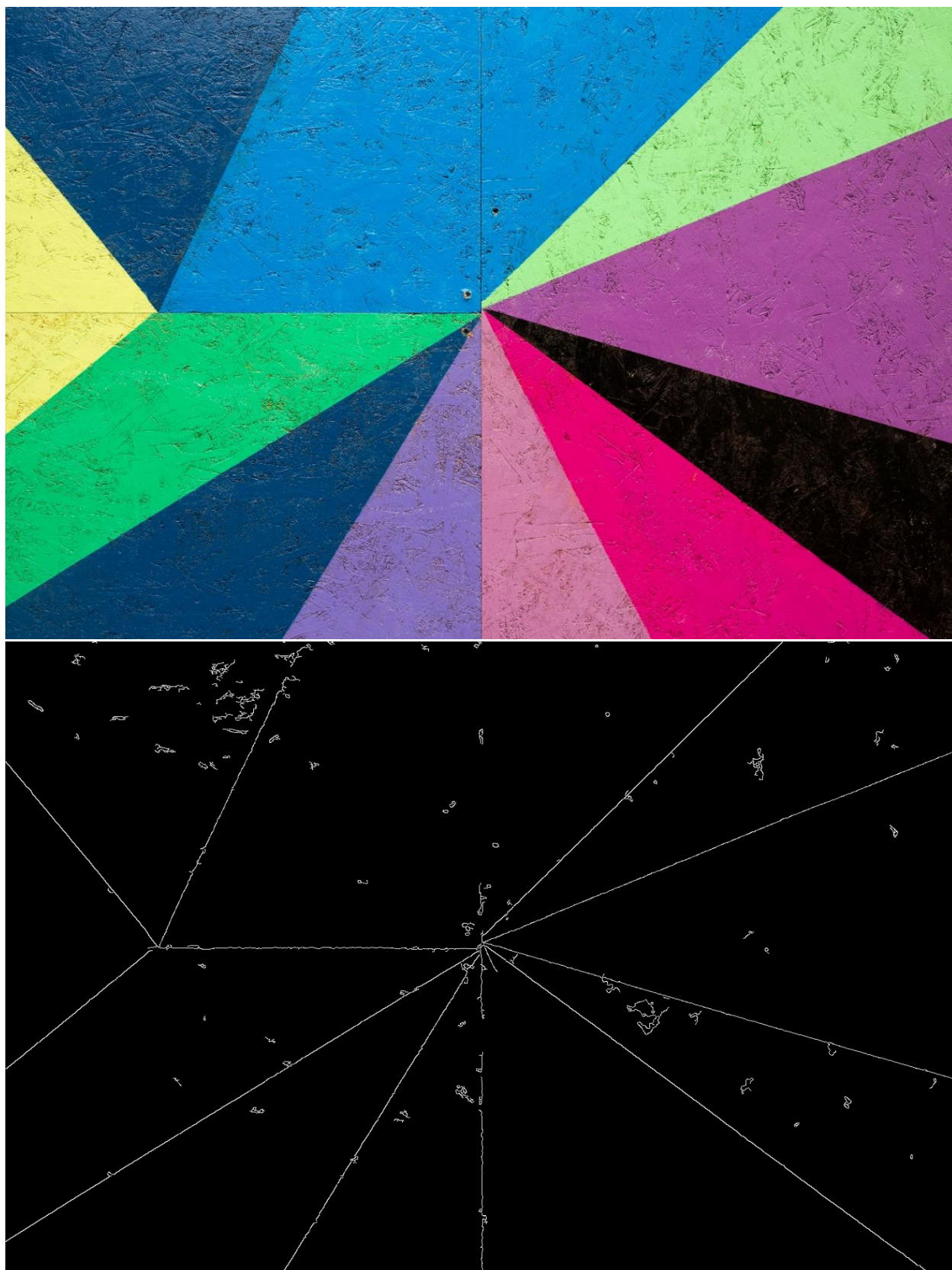


Рисунок 6 – пример работы оператора Лапласиана на изображение с четкими границами

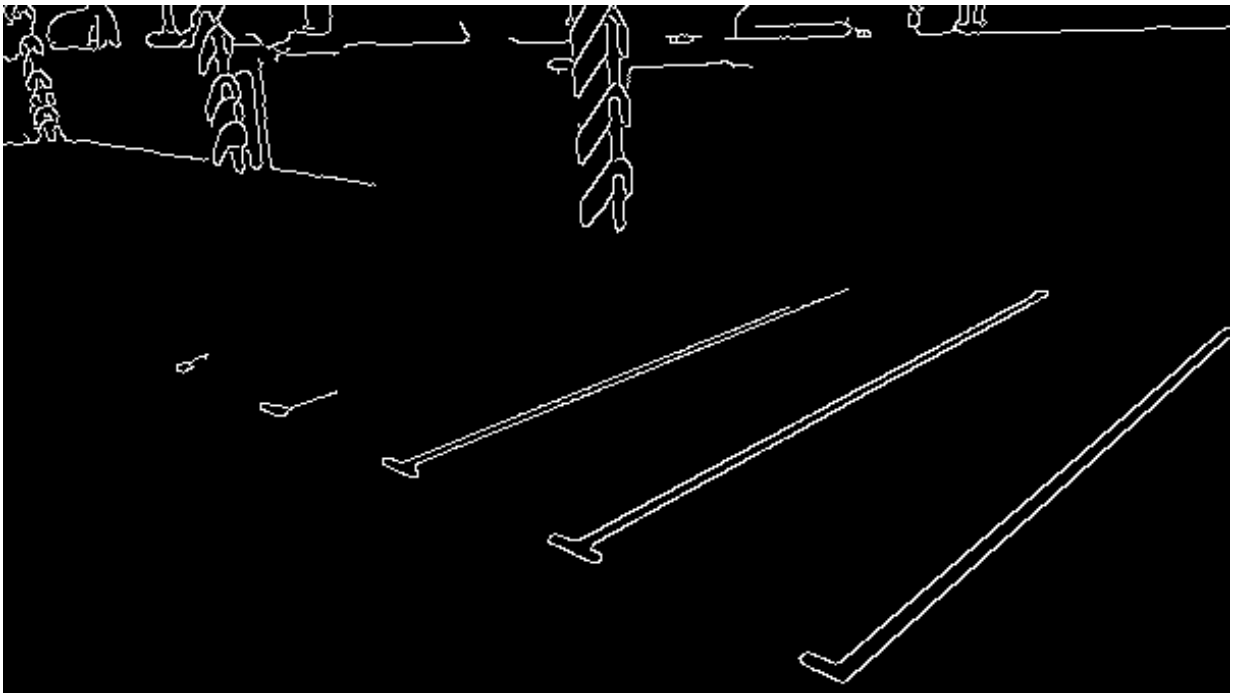


Рисунок 7 – пример работы оператора Лапласиана на изображение с нечеткими границами.

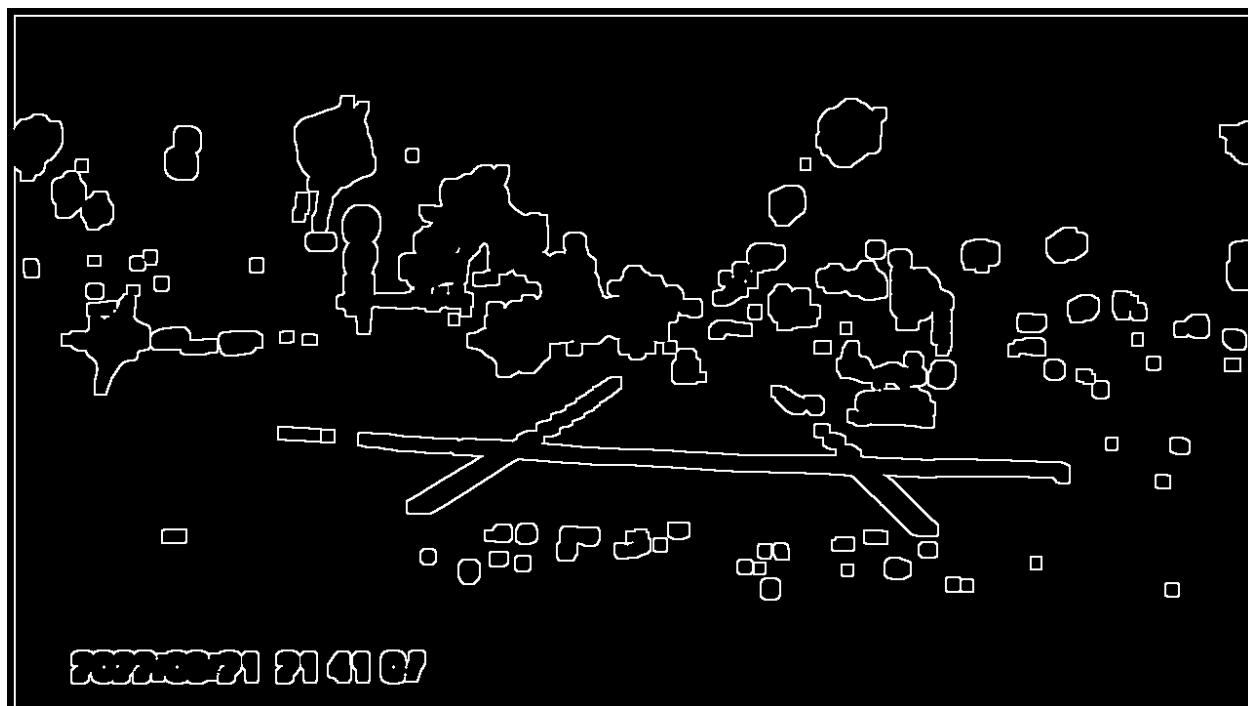


Рисунок 8 – пример работы оператора Лапласиана на видео в режиме реального времени

Вывод.

В ходе лабораторной работы были изучены методы выделения границ на изображение. Метода собеля ищет границы путем исследования производной по осям, что ограничивает нахождение вертикальных, горизонтальных и наклонных линий. Оператор Лапласиана позволяет искать границы на любом изображении.

Результаты работы

Код программы:

Задание 1

```
import cv2
import numpy

def nothing(x):
    pass

pathImg1 = "./lab5/5-1.jpg"
pathImg2 = "./lab5/5-2.jpg"
pathImg3 = "./lab5/5-3.jpg"

winName = "Sobel"
cv2.namedWindow(winName, cv2.WINDOW_GUI_NORMAL)

img = [cv2.imread(pathImg1, flags=cv2.IMREAD_COLOR), cv2.imread(pathImg2,
flags=cv2.IMREAD_COLOR), cv2.imread(pathImg3, flags=cv2.IMREAD_COLOR)]
ddepth = [cv2.CV_64F, cv2.CV_32F, cv2.CV_16F, cv2.CV_16S, cv2.CV_8U]

cv2.createTrackbar("xorder", winName, 0, 2, nothing)
cv2.createTrackbar("yorder", winName, 0, 2, nothing)
i = 0
while 1:

    xorder = cv2.getTrackbarPos("xorder", winName)
    yorder = cv2.getTrackbarPos("yorder", winName)
    if xorder+yorder==0: xorder=1
    grad_x = cv2.Sobel(img[i], ddepth=0, dx=xorder, dy=yorder, ksize=3, scale=1, delta=0,
borderType=cv2.BORDER_DEFAULT )
    cv2.imshow(winName, grad_x)
    key = cv2.waitKey(100)
    if key==27: break
    if key == 32: i=(i+1)%3

cv2.destroyAllWindows()
winName = "Laplacian"
cv2.namedWindow(winName, cv2.WINDOW_GUI_NORMAL)
while 1:

    dst = cv2.Laplacian(img[i], cv2.CV_8U, ksize=5, scale=1, delta=0,
borderType=cv2.BORDER_DEFAULT)
    cv2.imshow(winName, dst)
    key = cv2.waitKey(100)
    if key==27: break
    if key == 32: i=(i+1)%3
```

Задание 2

```
import cv2
import numpy

def nothing(x):
    pass

pathImg1 = "./lab5/5-4.PNG"
pathImg2 = "./lab5/5-5.jpg"

winName = "test_window"
cv2.namedWindow(winName, cv2.WINDOW_GUI_NORMAL)

img = [cv2.imread(pathImg1, flags=cv2.IMREAD_GRAYSCALE), cv2.imread(pathImg2,
flags=cv2.IMREAD_GRAYSCALE)]
ddepth = [cv2.CV_64F, cv2.CV_32F, cv2.CV_16F, cv2.CV_16S, cv2.CV_8U]

cv2.createTrackbar("threshold1", winName, 0, 100, nothing)
cv2.createTrackbar("threshold2", winName, 0, 200, nothing)

cv2.createTrackbar("size", winName, 0, 10, nothing)
# cv2.createTrackbar("iterations", winName, 0, 10, nothing)
morph_type = [cv2.MORPH_OPEN, cv2.MORPH_CLOSE,
               cv2.MORPH_GRADIENT, cv2.MORPH_TOPHAT, cv2.MORPH_BLACKHAT]

i = 0
while 1:

    threshold1 = cv2.getTrackbarPos("threshold1", winName)
    threshold2 = cv2.getTrackbarPos("threshold2", winName)
    size = 4 #cv2.getTrackbarPos("size", winName)

    size = cv2.getTrackbarPos("size", winName)+1
    iter_1 = 7 #cv2.getTrackbarPos("iterations", winName)+1

    t = 0

    # img_new = cv2.cvtColor(img[i], cv2.COLOR_BGR2GRAY)
    img_new = cv2.medianBlur(img[i], size*2+1)
    # threshold_new, img_new = cv2.threshold(img_new, 120, 255, cv2.THRESH_BINARY)

)

img_new = cv2.Canny(img_new, threshold1, threshold2)

cv2.imshow(winName, img_new)
key = cv2.waitKey(10)
if key==27:break
if key == 32: i=(i+1)%2

cv2.destroyAllWindows()
```

Задание 3

```
import cv2
import numpy

def nothing(x):
    pass

pathVideo1 = "./lab5/stop_line_1.mp4"
pathVideo2 = "./lab5/stop_line_2.mp4"

cap = cv2.VideoCapture(pathVideo1)

winName = "test_window"
cv2.namedWindow(winName, cv2.WINDOW_GUI_NORMAL)

cv2.createTrackbar("size", winName, 0, 10, nothing)
cv2.createTrackbar("iterations", winName, 0, 10, nothing)
# cv2.createTrackbar("type", winName, 0, 4, nothing)
i = 1
my_type = ["OPEN", "CLOSE"]
morph_type = [cv2.MORPH_OPEN, cv2.MORPH_CLOSE,
               cv2.MORPH_GRADIENT, cv2.MORPH_TOPHAT, cv2.MORPH_BLACKHAT]
p = 1
while 1:
    if p == 1:
        ret, frame = cap.read()

        threshold1 = 120 #cv2.getTrackbarPos("threshold1", winName)
        threshold2 = 250 #cv2.getTrackbarPos("threshold2", winName)
        size = 4 #cv2.getTrackbarPos("size", winName)
        # if frame is read correctly ret is True
        size_1 = 2 #cv2.getTrackbarPos("size", winName)+1
        iter_1 = 7 #cv2.getTrackbarPos("iterations", winName)+1

        size = 3 #cv2.getTrackbarPos("size", winName)+1
        iter = 0 #cv2.getTrackbarPos("iterations", winName)+1
        t = 0 #cv2.getTrackbarPos("type", winName)

        img_new = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        img_new = cv2.medianBlur(img_new, size*2+1)
        threshold_new, img_new = cv2.threshold(img_new, 120, 255, cv2.THRESH_BINARY)

        img_new = cv2.dilate(
            img_new,
            kernel=numpy.ones((size_1, size_1), dtype=int),
            # kernel = ker,
            anchor=(-1, -1),
            iterations=iter_1,
            borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255))

        img_new = cv2.morphologyEx(img_new, morph_type[t], kernel=numpy.ones((size, size),
            dtype=int),
            anchor=(-1, -1),
            iterations=iter,
            borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255),
            )

        img_new = cv2.Canny(img_new, threshold1, threshold2)
        # img_new = cv2.Laplacian(img_new, cv2.CV_8U, ksize=5, scale=1,delta=0,
        borderType=cv2.BORDER_DEFAULT)
        # img_new = cv2.Sobel(img_new, ddepth=0,dx=0, dy=1,ksize=3,scale=1, delta=0,
        borderType=cv2.BORDER_DEFAULT )

        cv2.imshow(winName, img_new)
```



```
key = cv2.waitKey(100)
if key == 32: p=1-p
if key == 27:
    break
if not ret:
    print("Can't receive frame (stream end?). Exiting ...")
    break

cap.release()
cv2.destroyAllWindows()
```