# Basics of line patrol: fast linear regression



## Introduction to the experimental results of the routine

> **[Fast linear regression]** is one of the commonly used strategies for intelligent car line patrol.

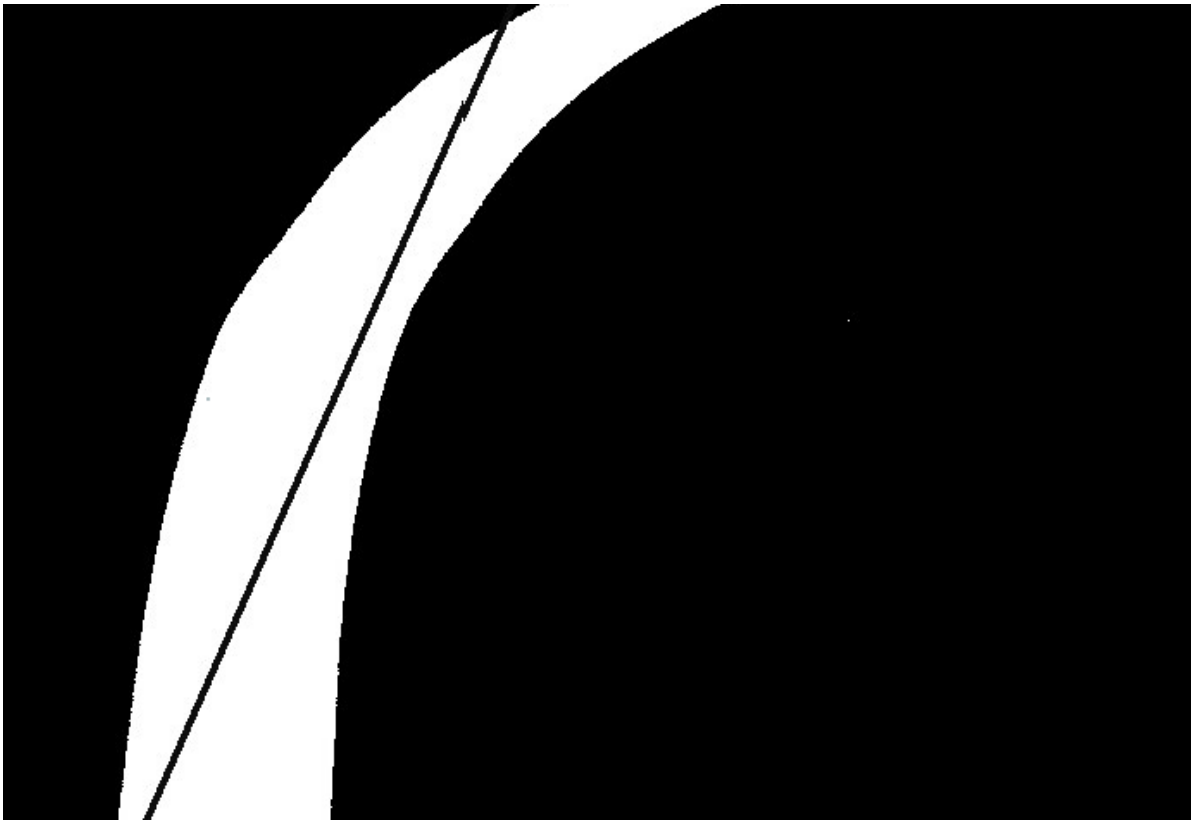
The routine code of this section is located at: [Source code summary / 04.Detecting / 05.linear_regression_fast.py]

We use CanMV IDE to open the routine code and connect K230 to the computer via USB

Click the run button in the lower left corner of CanMV IDE,

Aim the camera of K230 at the runway with [black line on white background]

You can see that there will be a black line on the screen. We regard the direction of this line as the direction in which the car should move when patrolling.



# Code explanation

The peripherals we will use in this section are mainly camera modules

Line segment detection is implemented by the get_regression() method in K230, which belongs to the image module

## Key code

```
# 初始化设备 / Initialize devices
sensor = init_sensor()
init_display()
sensor.run()

clock = time.clock()

# 计算显示偏移量以居中显示 / Calculate display offsets for center alignment
x_offset = round((DISPLAY_WIDTH - SENSOR_WIDTH) / 2)
y_offset = round((DISPLAY_HEIGHT - SENSOR_HEIGHT) / 2)

while True:
    clock.tick()

    # 捕获图像 / Capture image
    img = sensor.snapshot()

    # 处理图像 / Process image
    line = process_image(img)

    # 显示图像 / Display image
    Display.show_image(img, x=x_offset, y=y_offset)

    # 打印FPS和拟合度 / Print FPS and magnitude
    magnitude = str(line.magnitude()) if line else "N/A"
    print(f"FPS {clock.fps()}, mag = {magnitude}")
```

## Code structure

Image acquisition and processing:

- Capture grayscale images with a resolution of 640x480 through the camera
- You can choose whether to binarize the image (convert pixels with grayscale values in the range of 0-100 to white)

Linear regression detection:

- Perform linear regression analysis on the image to find the straight line features in the image
- Calculate the fit of the regression line (magnitude value), the larger the value, the closer it is to the straight line
- When a straight line is detected, draw this line on the image

Display function:

- Use the LCD screen to display the processed image

- The image is displayed in the center of the screen
- Real-time display of FPS (frames per second) and straight line fit

Program structure:

- Initialization part: Set camera parameters, display and media manager
- Main loop part: The process of continuously capturing images, processing and displaying
- Exception handling: Contains a complete exception handling mechanism and resource cleanup

# Fast Linear Regression Algorithm

## get_regression()

```
image.get_regression(thresholds[, invert=False[, roi[, x_stride=2[, y_stride=1[,
area_threshold=10[, pixels_threshold=10[, robust=False]]]]]]])
```

Compute linear regression on all thresholded pixels in the image. This is done using the least squares method, which is usually fast, but does not handle any outliers. If robust is True, the Theil index is used. The Theil index computes the median of all slopes between all thresholded pixels in the image. If you threshold too many pixels, this N^2 operation may drop your FPS below 5 even on an 80x60 image. However, as long as the number of pixels to threshold is small, linear regression will work even if more than 30% of the thresholded pixels are outliers.

This method returns an image.line object.

thresholds must be a list of tuples. [(lo, hi), (lo, hi), ..., (lo, hi)] defines the color range you want to track. For grayscale images, each tuple needs to contain two values - the minimum grayscale value and the maximum grayscale value. Only pixel regions that fall between these thresholds are considered. For RGB565 images, each tuple needs to have six values (l_lo, l_hi, a_lo, a_hi, b_lo, b_hi) - the minimum and maximum values of the LAB L, A and B channels, respectively. For ease of use, this function will automatically fix swapped minimum and maximum values. In addition, if the tuple is larger than six values, the remaining values are ignored. Conversely, if the tuple is too short, the remaining thresholds are assumed to be in the maximum range.

> Note: In addition to the fast linear regression algorithm, there are other feasible solutions for line inspection. We need to try to choose the best solution according to the actual situation.