Edge detection algorithm

1.Use

Code path：~/yahboomcar_ws/src/opencv_apps/launch

- Start the camera

```
roslaunch yahboomcar_visual opencv_apps.launch img_flip:=false
```

- img_flip parameter: whether the image needs to be flipped horizontally, the default is false.

[usb_cam-test.launch] file opens the [web_video_server] node by default, and you can directly use the [IP:8080] web page to view images in real time.

- Start the corner detection function of Opencv_apps

```
roslaunch opencv_apps edge_detection.launch              # Edge detection
algorithm
```

The debug_view for some functions is disabled, and there is no screen appearing. You can view the effect in the following two ways.

```
The reason for closing debug_view is that it will generate errors on the
terminal, but the actual effect has not been affected.
```

- Local View Screen

Enter the following command and select the corresponding topic to see the effect:

```
rqt_image_view
```

- LAN viewing screen

In the same local area network, enter IP+port (8080) in the browser, for example:

```
192.168.2.150:8080 # IP is the IP of the host computer
```

2. Effect display

There will be a topic for subscribing images and publishing images.

Parameter	Type	Default	Analyze
~use_camera_info	bool	true	Subscribe to the topic [camera_info] to get the default coordinate system ID, otherwise use the image information directly.
~debug_view	bool	false	Whether to create a window to display the node image
~edge_type	int	0	Specify the edge detection method: 0: Sobel operator, 1: Laplacian operator, 2: Canny edge detection
~canny_threshold1	int	100	Specify the second canny threshold
~canny_threshold2	int	200	Specify the first canny threshold
~apertureSize	int	3	Aperture size of the Sobel operator.
~apply_blur_pre	bool	True	Whether to apply blur() to the input image
~postBlurSize	double	3.2	Input image aperture size
~apply_blur_post	bool	False	Whether to apply GaussianBlur() to the input image
~L2gradient	bool	False	Canny's parameters
~queue_size	int	3	Queue size

Effect picture:

Node picture: