8.4、ROS+OpenCV application

8.4.1、Overview

wiki：http://wiki.ros.org/opencv_apps

Source code：https://github.com/ros-perception/opencv_apps.git

Code reference: https://github.com/Itseez/opencv/tree/master/samples/cpp

opencv_apps provides various nodes that run internally OpenCV's functionalities and publish the result as ROS topics. With opencv_apps, you can skip writing OpenCV application codes for a lot of its functionalities by simply running a launch file that corresponds to OpenCV's functionality you want.

There are related node analysis in ROS Wiki, topic subscription and topic publishing of corresponding nodes, introduction of related parameters, etc. Please check ROS WiKi for details.

8.4.2、Usage

1. Start up

Step 1: Start the camera

```
roslaunch jetbot_ros opencv_apps.launch
```

If the webpage cannot be viewed, check if there is no [web_video_server] node, if not, you need to run the following command.

```
rosrun web_video_server web_video_server
```

Step 2: Start the function of Opencv_apps

```
roslaunch opencv_apps corner_harris.launch                # harris corner
detection
roslaunch opencv_apps camshift.launch                     # Target tracking
algorithm
roslaunch opencv_apps contour_moments.launch              # Contour moment
roslaunch opencv_apps convex_hull.launch                  # Polygon outline
roslaunch opencv_apps find_contours.launch                # Contour detection
roslaunch opencv_apps general_contours.launch             # General contour
detection
roslaunch opencv_apps goodfeature_track.launch            # Feature point
tracking
roslaunch opencv_apps hough_circles.launch                # Hough circle
detection
roslaunch opencv_apps hough_lines.launch                  # Hough line detection
roslaunch opencv_apps lk_flow.launch                      # LK optical flow
algorithm
roslaunch opencv_apps phase_corr.launch                   # Phase correlation
displacement detection
roslaunch opencv_apps pyramids.launch                     # Image pyramid
sampling algorithm
roslaunch opencv_apps segment_objects.launch              # Clear background
detection algorithm
roslaunch opencv_apps smoothing.launch                    # Simple filter
```

Note: Some functions may need to modify the display resolution `xrandr --fb 640x480`, the default resolution is 640x480, the higher the resolution, the higher the vnc delay.

Almost every function case will have a parameter [debug_view], boolean type, whether to use Opencv to display the picture, it is displayed by default.

If you don't need to display it, set it to [False], for example

```
roslaunch opencv_apps contour_moments.launch debug_view:=False
```

After starting in this way, some cases can not be displayed in other ways, because in the source code, some [debug_view] is set to [False], and the image processing will be turned off.

2. Display method

- rqt_image_view

Enter the following command to select the corresponding topic

```
rqt_image_view
```

- opencv

The system displays it by default.

- Web view

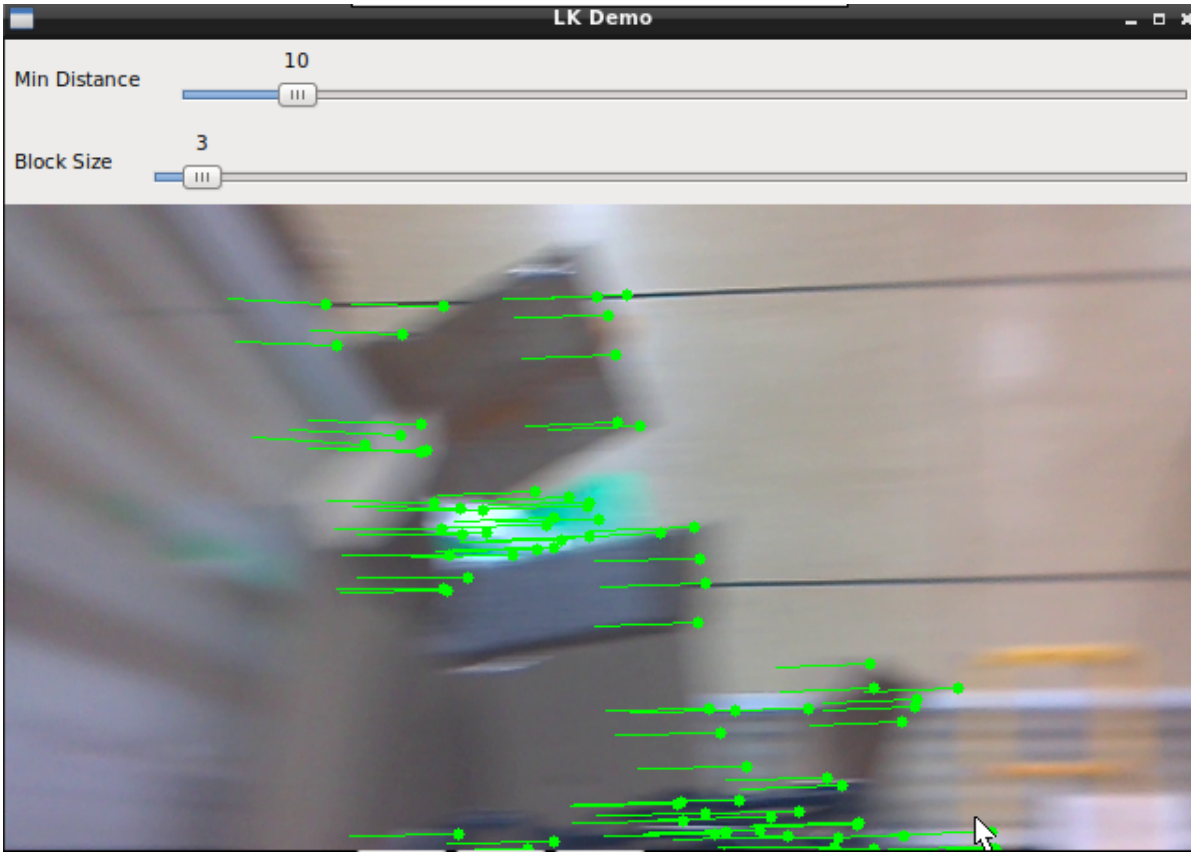
(At the same LAN) Enter IP+port in the browser, for example:

```
192.168.55.1:8080
```

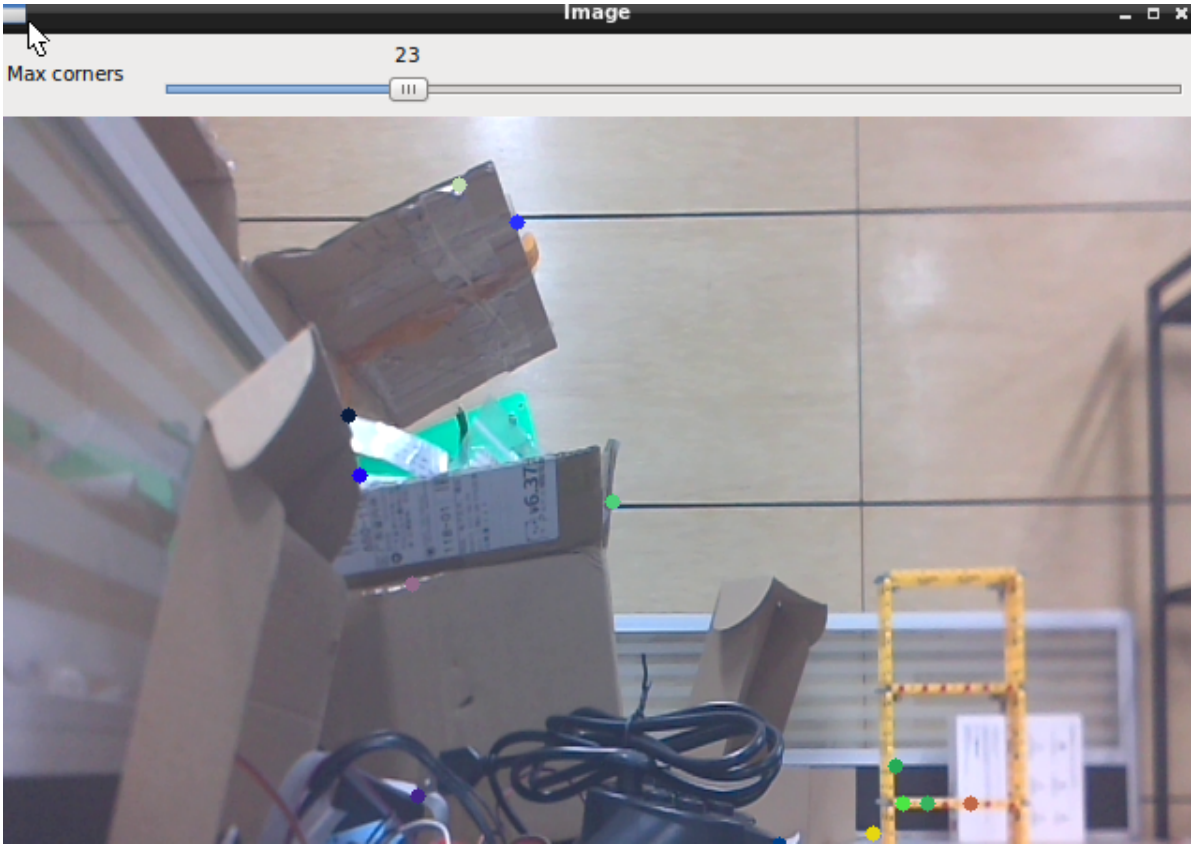
3. Effect show

- LK optical flow algorithm

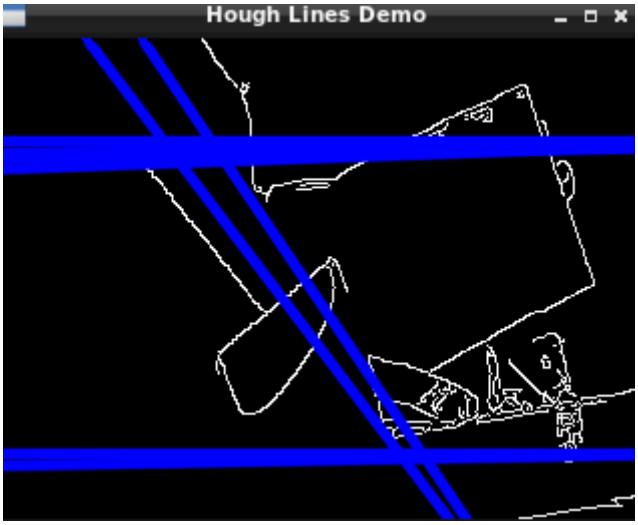
Move the screen and observe the phenomenon.



- Contour detection

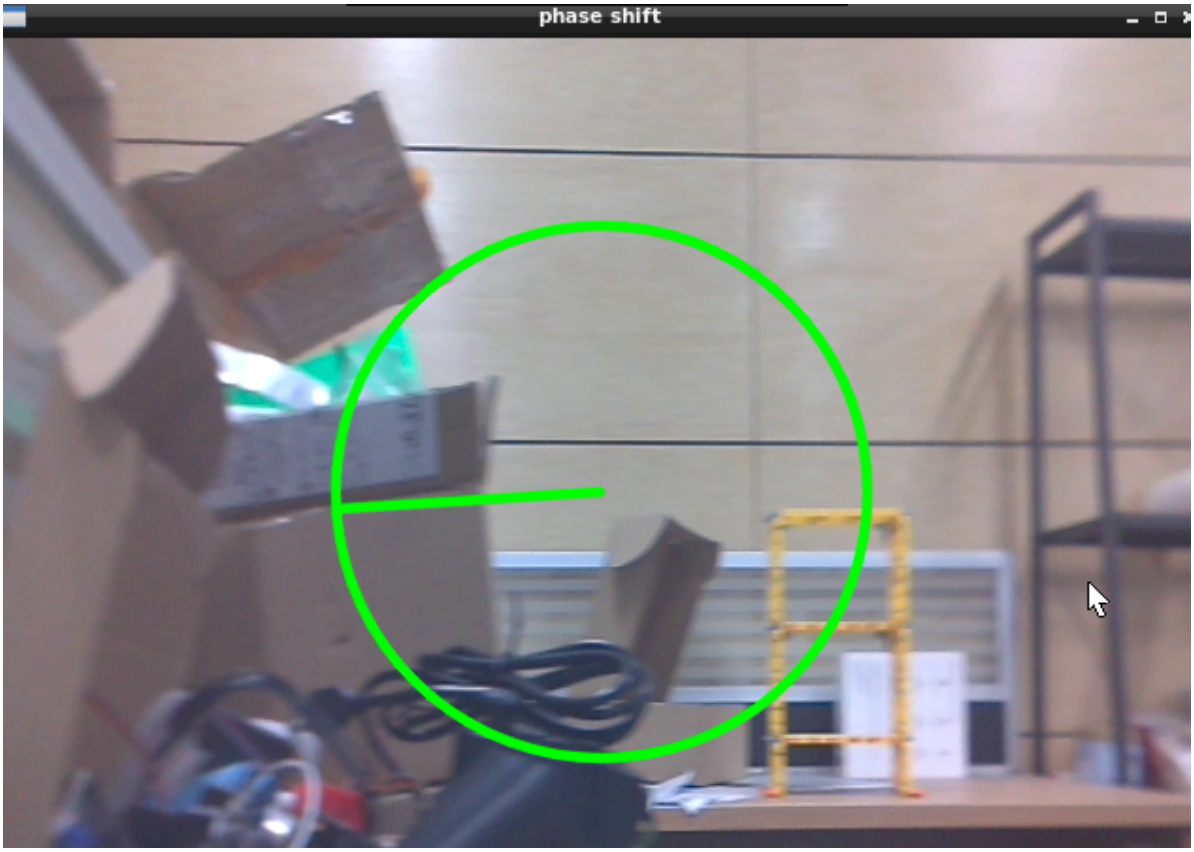


- Hough line detection



- Phase correlation displacement detection

The faster the camera moves, the larger the radius of the circle.



8.4.3 Node

In most cases, there will be a topic of subscribing to images and publishing images. Take the edge detection algorithmt as an example.

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
~canny_low_threshold	int	0	Canny edge detection low threshold
~queue_size	int	3	Queue size

code path: /home/jetson/workspace/catkin_ws/src/jetbot_ros/launch/opencv_apps.launch