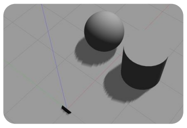
[Gazebo simulation] How to add binocular cameras to drones in gazebo

Unknown
Research monk

11 people agreed with the article

Add binocular camera

ROS Depth Camera Integration
🔗gazebosim.org



Gazebo plugins in ROS
🔗gazebosim.org

Build a depth camera model

In `px4/Firmware/Tools/sitl_gazebo/models` the new case of `kinect_self` the model in the catalog px4 new folder `kinect_self`
and create a new file in the folder `model.config` with the `model.sdf`
New `model.config` file

```
<?xml version="1.0"?>

<model>
  <name>Kinect_self</name>
  <version>1.0</version>
  <sdf version="1.5">model.sdf</sdf>
  <description>
        Work in progress.

        kinetic 3D camera
  </description>

</model>
```

New `model.sdf` file

```
<?xml version="1.0" ?>
<sdf version="1.5">
  <model name="kinect_self">
    <pose>-2 2.5 1 0 0 0</pose>
    <link name="link">
      <inertial>
        <mass>0.01</mass>
        <inertia>
          <ixx>4.16666666667e-06</ixx>
          <iyy>5.20833333333e-07</iyy>
          <izz>3.85416666667e-06</izz>
        </inertia>
      </inertial>

      <collision name="collision">
        <geometry>
          <box>
            <size>0.023000 0.076000 0.032000</size>
          </box>
        </geometry>
      </collision>
      <visual name="visual">
        <geometry>
          <box>
            <size>0.023000 0.076000 0.032000</size>
          </box>
        </geometry>
      </visual>
      <sensor type="depth" name="camera">
      <always_on>true</always_on>
      <update_rate>30.0</update_rate>
      <camera>
        <horizontal_fov>1.3962634</horizontal_fov>
        <image>
          <format>B8G8R8</format>
          <width>640</width>
          <height>480</height>
        </image>
        <clip>
          <near>0.4</near>
          <far>16.0</far>
        </clip>
      </camera>
      <plugin name="camera_plugin" filename="libgazebo_ros_openni_kinect.so">
        <baseline>0.2</baseline>
        <alwaysOn>true</alwaysOn>
        <!-- Keep this zero, update_rate in the parent <sensor> tag
          will control the frame rate. -->
        <updateRate>0.0</updateRate>
        <cameraName>camera_ir</cameraName>
        <imageTopicName>/camera/color/image_raw</imageTopicName>
        <cameraInfoTopicName>/camera/color/camera_info</cameraInfoTopicName>
        <depthImageTopicName>/camera/depth/image_raw</depthImageTopicName>
        <depthImageCameraInfoTopicName>/camera/depth/camera_info</depthImageCameraInfo
        <pointCloudTopicName>/camera/depth/points</pointCloudTopicName>
        <frameName>camera_link</frameName>
        <pointCloudCutoff>0.5</pointCloudCutoff>
        <pointCloudCutoffMax>8.0</pointCloudCutoffMax>
        <distortionK1>0</distortionK1>
        <distortionK2>0</distortionK2>
        <distortionK3>0</distortionK3>
        <distortionT1>0</distortionT1>
        <distortionT2>0</distortionT2>
        <CxPrime>0</CxPrime>
        <Cx>0</Cx>
        <Cy>0</Cy>
        <focalLength>0</focalLength>
        <hackBaseline>0</hackBaseline>
      </plugin>
    </sensor>
      <gravity>0</gravity>
    </link>
  </model>

</sdf>
```

This part mainly refers to the main reference and the depth camera model that comes with px4 `r200` (in `px4/Firmware/Tools/sitl_gazebo/models`)
where, modifying the parameters
`<pointCloudCutoff>0.5</pointCloudCutoff>` and

`<pointCloudCutoffMax>8.0</pointCloudCutoffMax>`

the value in can modify the working range of the depth camera

The depth camera model file is built, and then it needs to be added to the drone

Add depth camera

Opened `iris_fpv_cam` in `iris_fpv_cam.sdf`

```
<?xml version='1.0'?>
<sdf version='1.5'>
  <model name='iris_fpv_cam'>

    <include>
      <uri>model://iris</uri>
    </include>

    <!-- 前置双目 -->
    <include>
      <uri>model://kinect_self</uri>
      <pose>0.1 0 0 0 0 0</pose>
    </include>
    <joint name="kinect_joint" type="fixed">
      <child>kinect_self::link</child>
      <parent>iris::base_link</parent>
      <axis>
        <xyz>0 0 0</xyz>
        <limit>
          <upper>0</upper>
          <lower>0</lower>
        </limit>
      </axis>
    </joint>


    <!-- 下置摄像机  单目 -->
    <include>
      <uri>model://fpv_cam_down</uri>
      <pose>0 0 0 0 1.57 0</pose>
    </include>
    <joint name="fpv_cam_joint_down" type="fixed">
```

```
      <child>fpv_cam_down::link</child>
      <parent>iris::base_link</parent>
      <axis>
        <xyz>0 0 0</xyz>
        <limit>
          <upper>0</upper>
          <lower>0</lower>
        </limit>
      </axis>
    </joint>

  </model>
</sdf>
```

Modify `mavros_posix_sitl.launch` file

The file is `PX4/Firmware/launch` in, it is recommended to back up the original file before modifying

```
<?xml version="1.0"?>
<launch>
    <!-- MAVROS posix SITL environment launch script -->
    <!-- launches MAVROS, PX4 SITL, Gazebo environment, and spawns vehicle -->
    <!-- vehicle pose -->
    <arg name="x" default="0"/>
    <arg name="y" default="0"/>
    <arg name="z" default="0"/>
    <arg name="R" default="0"/>
    <arg name="P" default="0"/>
    <arg name="Y" default="0"/>
    <!-- vehicle model and world -->
    <arg name="est" default="ekf2"/>
    <arg name="vehicle" default="iris"/>
    <arg name="my_model" default="iris_fpv_cam"/>
    <arg name="world" default="$(find mavlink_sitl_gazebo)/worlds/empty.world"/>
<!--    <arg name="sdf" default="$(find mavlink_sitl_gazebo)/models/$(arg vehicle)/$(a
    <arg name="sdf" default="$(find mavlink_sitl_gazebo)/models/$(arg my_model)/$(arg

    <!-- gazebo configs -->
    <arg name="gui" default="true"/>
```

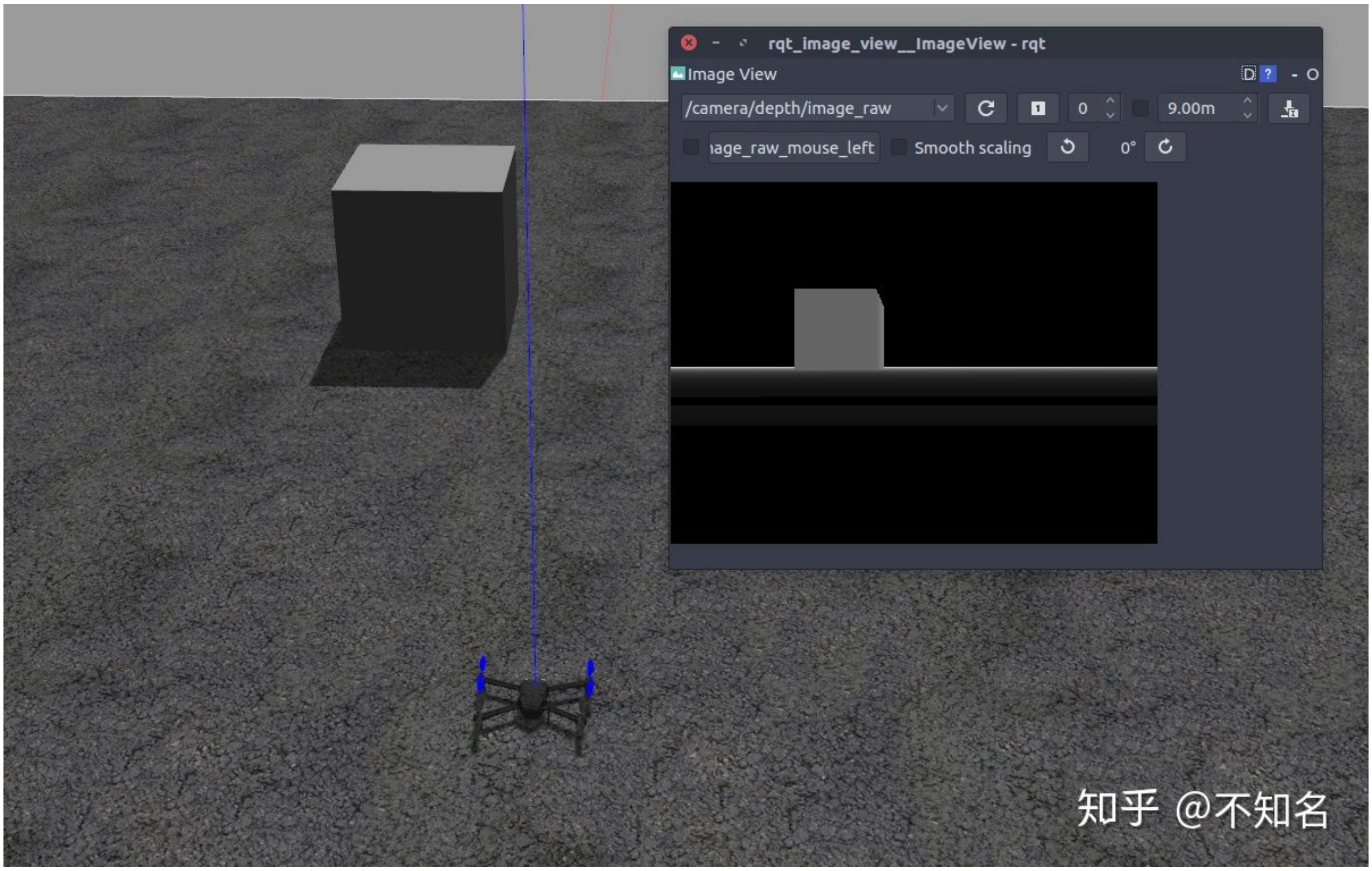
```
  <arg name="debug" default="false"/>
  <arg name="verbose" default="false"/>
  <arg name="paused" default="false"/>
  <arg name="respawn_gazebo" default="false"/>
  <!-- MAVROS configs -->
  <arg name="fcu_url" default="udp://:14540@localhost:14557"/>
  <arg name="respawn_mavros" default="false"/>
  <!-- PX4 configs -->
  <arg name="interactive" default="true"/>
  <!-- PX4 SITL and Gazebo -->
  <include file="$(find px4)/launch/posix_sitl.launch">
    <arg name="x" value="$(arg x)"/>
    <arg name="y" value="$(arg y)"/>
    <arg name="z" value="$(arg z)"/>
    <arg name="R" value="$(arg R)"/>
    <arg name="P" value="$(arg P)"/>
    <arg name="Y" value="$(arg Y)"/>
    <arg name="world" value="$(arg world)"/>
    <arg name="vehicle" value="$(arg vehicle)"/>
    <arg name="sdf" value="$(arg sdf)"/>
    <arg name="gui" value="$(arg gui)"/>
    <arg name="interactive" value="$(arg interactive)"/>
    <arg name="debug" value="$(arg debug)"/>
    <arg name="verbose" value="$(arg verbose)"/>
    <arg name="paused" value="$(arg paused)"/>
    <arg name="respawn_gazebo" value="$(arg respawn_gazebo)"/>
  </include>
  <!-- MAVROS -->
  <include file="$(find mavros)/launch/px4.launch">
    <!-- GCS link is provided by SITL -->
    <arg name="gcs_url" value=""/>
    <arg name="fcu_url" value="$(arg fcu_url)"/>
    <arg name="respawn_mavros" value="$(arg respawn_mavros)"/>
  </include>
</launch>
```

verification

Execution `launch` file

```
roslaunch px4 mavros_posix_sitl.launch
```

Add an object to the gazebo (the object faces the nose, the blue paddle is the nose)
In the new `terminal` performing `rqt_image_view` appears playing box (shown below on the right), for the selected topic `/camera/depth/image_raw` (which requires a depth camera model name `model.sdf` is `<depthImageTopicName>` consistent)


rqt_image_view__ImageView - rqt
Image View
/camera/depth/image_raw
0
9.00m
ıage_raw_mouse_left
Smooth scaling
0°
知乎 @不知名