ROS tutorial

navigation/sensor

ROBOT OPERATING SYSTEM LAB SESSION 6

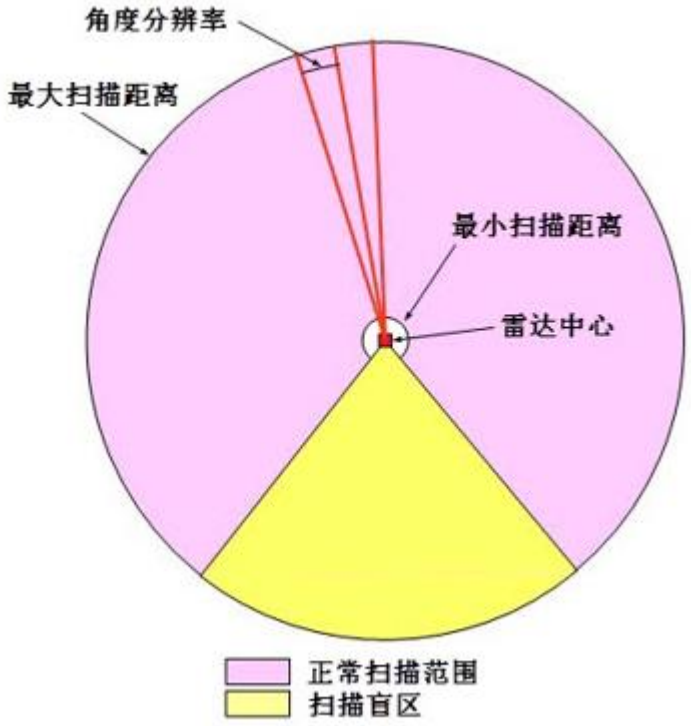
10/04/2018

Sensor message

Type:

- Image
- Imu
- Joy
- LaserScan
- PointCloud
- Temperature

Laser scan




```
# Single scan from a planar laser range-finder
#
# If you have another ranging device with different behavior (e.g. a sonar
# array), please find or create a different message, since applications
# will make fairly laser-specific assumptions about this data

Header header            # timestamp in the header is the acquisition time of
                         # the first ray in the scan.
                         #
                         # in frame frame_id, angles are measured around
                         # the positive Z axis (counterclockwise, if Z is up)
                         # with zero angle being forward along the x axis

float32 angle_min        # start angle of the scan [rad]
float32 angle_max        # end angle of the scan [rad]
float32 angle_increment  # angular distance between measurements [rad]

float32 time_increment   # time between measurements [seconds] - if your scanner
                         # is moving, this will be used in interpolating position
                         # of 3d points
float32 scan_time        # time between scans [seconds]

float32 range_min        # minimum range value [m]
float32 range_max        # maximum range value [m]

float32[] ranges         # range data [m] (Note: values < range_min or > range_max should be discarded)
float32[] intensities    # intensity data [device-specific units].  If your
                         # device does not provide intensities, please leave
                         # the array empty.
```

Source code

```
10  unsigned int num_readings = 100;
11  double laser_frequency = 40;
12  double ranges[num_readings];
13  double intensities[num_readings];
```

```
19  for(unsigned int i = 0; i < num_readings; ++i){
20    ranges[i] = count;
21    intensities[i] = 100 + count;
22  }
23  ros::Time scan_time = ros::Time::now();
```

Source code

```
26  sensor_msgs::LaserScan scan;
27  scan.header.stamp = scan_time;
28  scan.header.frame_id = "laser_frame";
29  scan.angle_min = -1.57;
30  scan.angle_max = 1.57;
31  scan.angle_increment = 3.14 / num_readings;
32  scan.time_increment = (1 / laser_frequency) / (num_readings);
33  scan.range_min = 0.0;
34  scan.range_max = 100.0;
36  scan.ranges.resize(num_readings);
37  scan.intensities.resize(num_readings);
38  for(unsigned int i = 0; i < num_readings; ++i){
39    scan.ranges[i] = ranges[i];
40    scan.intensities[i] = intensities[i];
41  }
```

test

1 Publishing Odometry Information over ROS

http://wiki.ros.org/navigation/Tutorials/RobotSetup/Odom

2 follow the example of laser and odometry, Publishing camera Information over ROS

File: sensor_msgs/Image.msg

Raw Message Definition

```
# This message contains an uncompressed image
# (0, 0) is at top-left corner of image
#

Header header        # Header timestamp should be acquisition time of image
                     # Header frame_id should be optical frame of camera
                     # origin of frame should be optical center of cameara
                     # +x should point to the right in the image
                     # +y should point down in the image
                     # +z should point into to plane of the image
                     # If the frame_id here and the frame_id of the CameraInfo
                     # message associated with the image conflict
                     # the behavior is undefined

uint32 height         # image height, that is, number of rows
uint32 width          # image width, that is, number of columns

# The legal values for encoding are in file src/image_encodings.cpp
# If you want to standardize a new string format, join
# ros-users@lists.sourceforge.net and send an email proposing a new encoding.

string encoding       # Encoding of pixels -- channel meaning, ordering, size
                      # taken from the list of strings in include/sensor_msgs/image_encodings.h

uint8 is_bigendian    # is this data bigendian?
uint32 step           # Full row length in bytes
uint8[] data          # actual matrix data, size is (step * rows)
```

Compact Message Definition

```
std_msgs/Header header
uint32 height
uint32 width
string encoding
uint8 is_bigendian
uint32 step
uint8[] data
```