How to convert a cv::Mat into a sensor_msgs in ros?

Ask Question

I am trying to convert a cv::Mat into a sensor_msgs so that I can publish this in ROS.

my code is just like this :

```
while(ros::ok())
        {
                capture >> frame;
                cv::imshow("Preview" , frame);
                cv::waitKey(1);
                //sensor_msgs::Image img_;
                //fillImage(img_ , "rgb8" , frame.rows , frame.cols , 3 *
frame.cols , frame);
                //img_header.stamp = ros::Time::now();
                //cv_bridge::CvImagePtr cv_ptr;
                //cv_ptr->image = frame;
                //image_pub_.publish(img_);
                ros::spinOnce();
        }
```

I have tried two potential solutions :

[1] using cv_bridge, CvImagePtr and toImageMsg(), but the CvImagePtr report

assert(px!0) error, which I guess means that I have to initialize CvImagePtr.

But I don't know how to initialize it;

[2] using fillImage and sensor_msgs::Image,

but the sixth parameter of fillImage has to be a void* instead of a Mat*

Hope anyone could help me !

Is there an efficient way to convert cv::Mat(or IplImage) to sensor_msgs ?

THX in advance !

c++ pointers opencv type-conversion ros

asked Nov 22 '14 at 17:05
Thu Vlion
31 1 5

Have a look here: wiki.ros.org/cv_bridge/Tutorials/... (and here: answers.ros.org/question/9765/...) – alextoind Nov 23 '14 at 10:29

thx alex, the second link helps alot ! – Thu Vlion Nov 24 '14 at 3:38

2 Answers

- Using image_transport: http://wiki.ros.org/image_transport/Tutorials/PublishingImages
- Using cv_bridge interface and setting the header by hand

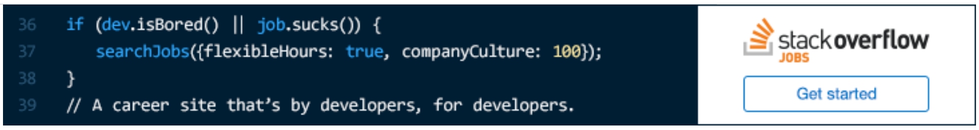
Use the following code

```
#include <sensor_msgs/Image.h>
#include <sensor_msgs/image_encodings.h>
#include <cv_bridge/cv_bridge.h>

cv::Mat img; // << image MUST be contained here
cv_bridge::CvImage img_bridge;
sensor_msgs::Image img_msg; // >> message to be sent

std_msgs::Header header; // empty header
header.seq = counter; // user defined counter
header.stamp = ros::Time::now(); // time
img_bridge = cv_bridge::CvImage(header, sensor_msgs::image_encodings::RGB8, img);
img_bridge.toImageMsg(img_msg); // from cv_bridge to sensor_msgs::Image
pub_img.publish(img_msg); // ros::Publisher pub_img =
node.advertise<sensor_msgs::Image>("topic", queuesize);
```

answered Jan 5 '16 at 17:44



I think the best shot for you is to read the ROS tutorial Converting between ROS images and OpenCV images (C++):

> This tutorial describes how to interface ROS and OpenCV by converting ROS images into OpenCV images, and vice versa, using cv_bridge. Included is a sample node that can be used as a template for your own node.

answered 2 days ago

jirislav

130 1 11