```cpp
#include <ros/ros.h>
#include <image_transport/image_transport.h> //用来发布和订阅图像信息
#include <opencv2/highgui.hpp>
#include <opencv2/calib3d.hpp>
#include <cv_bridge/cv_bridge.h> //将 ROS 下的 sensor_msgs/Image 消息类型转换为 cv::Mat 数据类型
#include <iostream>
//···程序中注释掉的代码为直接运用 cv 提供的 gui 窗口显示相机数据
int main(int argc, char** argv)
{
  ros::init(argc, argv, "camera_pub_node");  //ROS 初始化，定义节点名称 camera_pub_node
  ros::NodeHandle nh;                        //定义 ROS 句柄
  image_transport::ImageTransport it(nh);  //实例化句柄
  image_transport::Publisher pub = it.advertise("camera/image", 1);  //发布话题名
/camera/image

  cv::Mat frame;   //原始图像保存
  cv::VideoCapture capture(2);   // 创建摄像头捕获，并打开摄像头 2(一般是 0,2....)

  if(capture.isOpened() == 0)      // 如果摄像头没有打开
  {
    std::cout << "Read camera failed!" << std::endl;
    return -1;
  }
 else
  {
   std::cout << "Read camera successful!" << std::endl;
  }
  //cv::namedWindow("view");  //命名一个新窗口
  //cv::startWindowThread();   //运行一个新的窗口
  ros::Rate loop_rate(30); //设置发布频率为 30Hz
  while (nh.ok())
  {
    capture.read(frame);  //读取当前图像到 frame
    //cv::imshow("view",frame);  //在窗口中显示 frame 内容，这是直接的 cv::Mat 的格式
    if(!frame.empty())
    {
      sensor_msgs::ImagePtr msg = cv_bridge::CvImage(std_msgs::Header(), "bgr8",
frame).toImageMsg(); //运用 cv_brigde 将图像格式转换成 ROS 格式
      pub.publish(msg);  //发布消息
    }
    ros::spinOnce();
    loop_rate.sleep(); //照应发布频率
  }
  //cv::destroyWindow("view"); //关闭窗口
  return 0;
}
```