Contact

Domenico Daniele Bloisi, PhD
Assistant Professor

Location: DIAG A209

bloisi@dis.uniroma1.it
http://www.dis.uniroma1.it/~bloisi
Introduction
Microsoft Kinect

- Kinect was launched in North America on November 4th 2010
- Kinect is a motion sensing input device
Microsoft Kinect

- Depth Sensors
- RGB Camera
- Motorized Tilt
- Multi-array Mic

Sensor Data Acquisition

Robot Programming
Depth Sensors

Consists of IR projector and a CMOS sensor

IR beam bounces off subject and is captured by the CMOS sensor

Sensor uses time to measure distance of objects
Depth Sensors

RGB Camera

640 x 480 pixels at 30 frames per second (fps)

11-bit depth
RGB + Depth

http://graphics.stanford.edu/~mdfisher/Kinect.html
Camera Physical Limitations

Angular field of view:
57° horizontally
4.0 m
0.8 m

Angular field of view:
43° vertically
Tilt up to 27° up or down
+27°
43°
-27°
4.0 m
Microphones

4 Microphones on the device

Supports single speaker voice recognition

16-bit audio sampled at 16kHz
Visualizing RGBD Data

Sensor Data Acquisition

Robot Programming
OpenCV (Open Source Computer Vision) is a library of programming functions for realtime computer vision.

- BSD Licensed - free for commercial use
- C++, C, Python and Java (Android) interfaces
- Supports Windows, Linux, Android, iOS and Mac OS
- More than 2500 optimized algorithms
Libraries

//ROS
#include <ros/ros.h>
#include <image_transport/image_transport.h>
#include <cv_bridge/cv_bridge.h>
#include <sensor_msgs/image_encodings.h>

//OpenCV
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
main

```c
int main(int argc, char **argv) {
    ros::init(argc, argv, "kinectgrabber");
    ros::NodeHandle n;
    ros::Subscriber sub =
        n.subscribe("/camera/rgb/image_color", 1, rgbCallback);
    ros::Subscriber depth =
        n.subscribe("/camera/depth/image", 1, depthCallback);
    ros::spin();
    return 0;
}
```
void rgbCallback(const sensor_msgs::ImageConstPtr& msg)
{
    cv_bridge::CvImageConstPtr cv_ptr;
    try
    {
        cv_ptr = cv_bridge::toCvCopy(msg, msg->encoding);
    }
    catch (cv_bridge::Exception& ex)
    {
        ROS_ERROR("cv_bridge exception: %s", ex.what());
        exit(-1);
    }
    cv::imshow("RGB", cv_ptr->image);
    cv::waitKey(30);
}
void depthCallback(const sensor_msgs::ImageConstPtr& msg)
{
    cv_bridge::CvImageConstPtr cv_ptr;
    try{
        cv_ptr = cv_bridge::toCvCopy(msg, msg->encoding);
    }
    catch (cv_bridge::Exception& ex)
    {
        ROS_ERROR("cv_bridge exception: %s", ex.what());
        exit(-1);
    }

    cv::imshow("DEPTH", cv_ptr->image);
    cv::waitKey(30);
}
Result

Sensor Data Acquisition

Robot Programming
Source code

http://www.dis.uniroma1.it/~bloisi/didattica/RobotProgramming/kinectGrabber.zip
The Point Cloud Library (PCL) is a standalone, large scale, open project for 2D/3D image and point cloud processing.

- Collection of Libraries focused on Point Cloud processing
- More than 450 developers/contributors
- Over 60 Tutorials and many examples
- BSD Licensed - free for commercial use
Visualizing 3D Data

Sensor Data Acquisition

Robot Programming
Point Cloud
Point Cloud

Perception with RGBD sensor- 3D, November 19th 3D data processing