# OpenNI2与OpenCV2结合显示深度图像

以下程序示例了在OpenNI2中如何获取深度图像并使用OpenCV显示出来：

#include "stdafx.h"

#include <opencv2\opencv.hpp>

#include <OpenNI.h>

using namespace std;

using namespace cv;

using namespace openni;

bool HandleStatus(openni::Status status)

{

if (status == STATUS\_OK)

return true;

std::cout << "ERROR:" << OpenNI::getExtendedError() << std::endl;

return false;

}

void ConvertDetphToGrayMat(VideoFrameRef& frame, int maxDepth, Mat& gray)

{

if (!frame.isValid()) return;

gray.create(Size(frame.getWidth(), frame.getHeight()), CV\_8UC1);

for (int i = 0; i < gray.rows; i++)

{

char\* pdata = gray.ptr<char>(i);

DepthPixel\* pdeth = (DepthPixel\*)((char\*)frame.getData()+i \* frame.getStrideInBytes());

for (int j = 0; j < gray.cols; j++)

{

if (pdeth[j] != 0)

pdata[j] = 255 - pdeth[j] \* 255 / maxDepth;

else

pdata[j] = 0;

}

}

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

Status status = STATUS\_OK;

Device device;

status = OpenNI::initialize();

if (!HandleStatus(status))return 1;

status = device.open(ANY\_DEVICE);

if (!HandleStatus(status))return 1;

VideoStream depth;

status = depth.create(device, SENSOR\_DEPTH);

if (!HandleStatus(status))return 1;

VideoStream\* pSensor = &depth;

VideoMode vm;

vm.setFps(30);

vm.setResolution(640, 480);

vm.setPixelFormat(PIXEL\_FORMAT\_DEPTH\_1\_MM);

status = depth.setVideoMode(vm);

if (!HandleStatus(status))return 1;

Mat depthMat(Size(640,480),CV\_8UC1,Scalar(0));

depth.start();

int streamIndex = 0;

while (1)

{

//获取深度数据

status = OpenNI::waitForAnyStream(&pSensor, 1, &streamIndex);

if (!HandleStatus(status))return 1;

VideoFrameRef frame;

status = pSensor->readFrame(&frame);

if (HandleStatus(status) && frame.isValid())

{

//最大深度

int iMaxDepth = pSensor->getMaxPixelValue();

Mat gray;

//将深度图像转化为OPENCV的灰度图像

ConvertDetphToGrayMat(frame, iMaxDepth, gray);

if (gray.data)

{

equalizeHist(gray, gray);

imshow("depth", gray);

}

char key = waitKey(10);

if (key == 27)

{

break;

}

}

}

depth.stop();

device.close();

OpenNI::shutdown();

return 0;

}

