Using OpenCV with Qt

Introduction

In this article we will see how to use OpenCV in Qt.

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real time computer vision, developed by Intel and now supported by Willow Garage. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on *real-time* image processing. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.

The IplImage and Qt

In order to take advantage of the 500 algorithms offered by OpenCV, Qt applications have to manage **iplImages**. Therefore, converting `QImage` to/from **iplImage** is very important.

Converting QImage to iplImage

This snippet will convert `QImages` into **iplImage** with depth 8 and 3 channels. `QImage` can store data in several formats. This code accepts just 24-bit **QImage::Format_RGB888** and the **QImage::Format_RGB32**. Apha values in the 32-bit format will be removed during the conversion.

```
static IplImage* qImage2IplImage(const QImage& qImage)
{
  int width = qImage.width();
  int height = qImage.height();

  // Creates a iplImage with 3 channels
  IplImage *img = cvCreateImage(cvSize(width, height), IPL_DEPTH_8U, 3);
  char * imgBuffer = img->imageData;

  //Remove alpha channel
  int jump = (qImage.hasAlphaChannel()) ? 4 : 3;

  for (int y=0;y<img->height;y++){
    QByteArray a((const char*)qImage.scanLine(y), qImage.bytesPerLine());
    for (int i=0; i<a.size(); i+=jump){
        //Swap from RGB to BGR
        imgBuffer[2] = a[i];
        imgBuffer[1] = a[i+1];
        imgBuffer[0] = a[i+2];
        imgBuffer+=3;
    }
}

return img;
}
```

Converting iplImage to QImage

This snippet will convert a **iplImage** with depth 8 and 1 or 3 channels into a 8/24-bit `QImage`.

Note: This code won't work for images with different depth and number of channels.

```
static QImage IplImage2QImage(const IplImage *iplImage)
{
    int height = iplImage->height;
    int width = iplImage->width;

    if  (iplImage->depth == IPL_DEPTH_8U && iplImage->nChannels == 3)
    {
      const uchar *qImageBuffer = (const uchar*)iplImage->imageData;
      QImage img(qImageBuffer, width, height, QImage::Format_RGB888);
      return img.rgbSwapped();
    } else if  (iplImage->depth == IPL_DEPTH_8U && iplImage->nChannels == 1){
 const uchar *qImageBuffer = (const uchar*)iplImage->imageData;
 QImage img(qImageBuffer, width, height, QImage::Format_Indexed8);

 QVector<QRgb> colorTable;
 for (int i = 0; i < 256; i++){
     colorTable.push_back(qRgb(i, i, i));
 }
 img.setColorTable(colorTable);
 return img;
    }else{
      qWarning() << "Image cannot be converted.";
      return QImage();
    }
}
```

Useful Links

http://opencv.willowgarage.com/wiki/
http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/opencv-intro.html
http://doc.qt.nokia.com/4.6/qimage.html