<http://www.raspberry-projects.com/pi/category/programming-in-c>

<http://www.raspberry-projects.com/pi/programming-in-c/getting-your-raspberry-pi-ready-for-c-programming>

--Programming in C++ on the Raspberry Pi, includes links to compilers

[https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/image\_processing/https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/image\_processing/](https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/image_processing/)

-- Custom libraries and simple explanations for webcam and raspberry pi

<http://programmaticponderings.wordpress.com/2013/02/09/opencv-and-cvblob-with-raspberry-pi/>

--Using OpenCV on the Raspberry PI

<http://blog.ayoungprogrammer.com/2013/07/tutorial-scanning-barcodes-qr-codes.html>

<http://blog.ayoungprogrammer.com/2014/04/real-time-qr-code-bar-code-detection.html>

--Tutorials using Zbar and OpenCV for barcode/QR code detections from a video feed.

--Working example of QR code detection using OpenCV and Zbar

--Is in C++, may need to convert to PYTHON  
#include <opencv2/highgui/highgui.hpp>   
 #include <opencv2/imgproc/imgproc.hpp>   
 #include <zbar.h>   
 #include <iostream>   
 using namespace cv;   
 using namespace std;   
 using namespace zbar;   
 //g++ main.cpp /usr/local/include/ /usr/local/lib/ -lopencv\_highgui.2.4.8 -lopencv\_core.2.4.8   
 int main(int argc, char\* argv[])   
 {   
 VideoCapture cap(0); // open the video camera no. 0   
 // cap.set(CV\_CAP\_PROP\_FRAME\_WIDTH,800);   
 // cap.set(CV\_CAP\_PROP\_FRAME\_HEIGHT,640);   
 if (!cap.isOpened()) // if not success, exit program   
 {   
 cout << "Cannot open the video cam" << endl;   
 return -1;   
 }   
 ImageScanner scanner;   
 scanner.set\_config(ZBAR\_NONE, ZBAR\_CFG\_ENABLE, 1);   
 double dWidth = cap.get(CV\_CAP\_PROP\_FRAME\_WIDTH); //get the width of frames of the video   
 double dHeight = cap.get(CV\_CAP\_PROP\_FRAME\_HEIGHT); //get the height of frames of the video   
 cout << "Frame size : " << dWidth << " x " << dHeight << endl;   
 namedWindow("MyVideo",CV\_WINDOW\_AUTOSIZE); //create a window called "MyVideo"   
 while (1)   
 {   
 Mat frame;   
 bool bSuccess = cap.read(frame); // read a new frame from video   
 if (!bSuccess) //if not success, break loop   
 {   
 cout << "Cannot read a frame from video stream" << endl;   
 break;   
 }   
 Mat grey;   
 cvtColor(frame,grey,CV\_BGR2GRAY);   
 int width = frame.cols;   
 int height = frame.rows;   
 uchar \*raw = (uchar \*)grey.data;   
 // wrap image data   
 Image image(width, height, "Y800", raw, width \* height);   
 // scan the image for barcodes   
 int n = scanner.scan(image);   
 // extract results   
 for(Image::SymbolIterator symbol = image.symbol\_begin();   
 symbol != image.symbol\_end();   
 ++symbol) {   
 vector<Point> vp;   
 // do something useful with results   
 cout << "decoded " << symbol->get\_type\_name() << " symbol \"" << symbol->get\_data() << '"' <<" "<< endl;   
 int n = symbol->get\_location\_size();   
 for(int i=0;i<n;i++){   
 vp.push\_back(Point(symbol->get\_location\_x(i),symbol->get\_location\_y(i)));   
 }   
 RotatedRect r = minAreaRect(vp);   
 Point2f pts[4];   
 r.points(pts);   
 for(int i=0;i<4;i++){   
 line(frame,pts[i],pts[(i+1)%4],Scalar(255,0,0),3);   
 }   
 //cout<<"Angle: "<<r.angle<<endl;   
 }   
 imshow("MyVideo", frame); //show the frame in "MyVideo" window   
 if (waitKey(30) == 27) //wait for 'esc' key press for 30ms. If 'esc' key is pressed, break loop   
 {   
 cout << "esc key is pressed by user" << endl;   
 break;   
 }   
 }   
 return 0;   
 }

<http://zbar.sourceforge.net/> -- For reading the QR code

--Live reading of video streams