

SHANGHAI JIAOTONG UNIVERSITY

VISUAL SCENE REAL-TIME ANALYSIS

Automatic QR Code Recognition

Author:

LI Yanhao
118260910036

Supervisor:

Hao LI

November 14, 2019







1 Task description

Nowadays QR code is in common use everywhere in our life. Automatic QR code recognition is currently a mature technique. Our task is to implement an algorithm to recognize a QR code and output a binary matrix. The QR code size is 42x42, thus the binary matrix is of size 42x42 with values equal to 1 for white and 0 for black.

2 Solution

To extract a regular binary matrix, we need a QR code in a regular shape, ideally a square. However, the scanned QR code is usually distorted due to an irregular scanning direction. We need to first reform the scanned QR code in a square shape, then identify the color of each small square and output 0 or 1. The solution is divided in the steps as follows:

- Read the QR code image and convert it to a gray scale image.
- Identify three boxes in the shape of . The shape of  contains three successive contours from outside to inside. We utilize findContour() function of opencv library to identify the contours that encircle a second contour, with the latter encircling a third contour.
- Determine the position of each box of . We have extracted three boxes in the previous step, but still don't know which position each box belongs to. We want to identify the top left, bottom left and top right boxes. Since the three boxes are located on three summits of a straight triangle if the QR code is a square, in most cases the side corresponding to the top left box after distortion is still the longest side. We thus find out the longest side of the triangle, and also its corresponding top left box. Then we find out the rest of two boxes by calculating the cross product of their corresponding side vectors.
- For each box of , extract its four corner points using the opencv function goodFeaturesToTrack(). Then identify the positions of the four corner points according to their relative positions to the other two boxes.
- So far, we have identified 12 corner points of 3 boxes in the scanned QR code image, and we know the exact positions of these 12 points in a regular square QR code. We use the findHomography() function to find out the homography matrix between the distorted QR code and regular QR code.
- Reform the QR code using warpPerspective() function with the obtained homography matrix. Define a grid of size 42x42 on the square QR code.

Calculate the mean value of pixels for each small square. If the mean value is superior to 128, the small square is considered white, else considered black. Set the corresponding element of the output matrix to 1 or 0.

- Write the output matrix to a file.

3 Evaluation

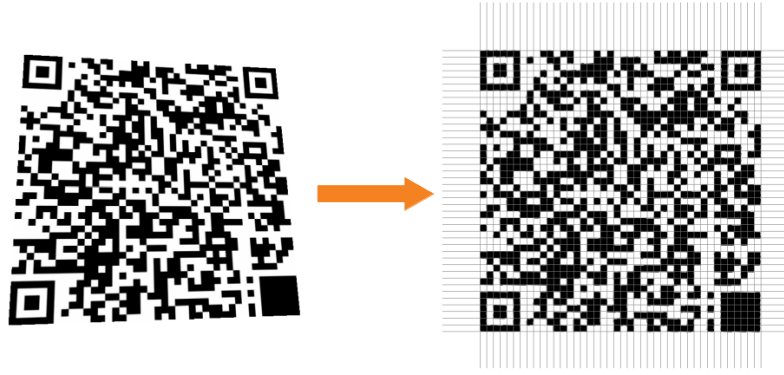


Figure 1: Homography transformation

4 Demonstration

5 Example

The feasibility of this LRC code is tested in the hadoop file system in the following steps:

- Compile the whole hadoop project, package it into a native distribution and deploy the hadoop system using docker container.
- Create an empty directory in hadoop file system. Set the erasure code policy to LRC-6-4-1024k.
- Put an 800 MB file in the directory. The file will be encoded by LRC code into 6 data chunks and 4 parity chunks.
- Find the physical locations of file chunks. Choose several chunks and manually delete them.

- Fetch the file from the hadoop file system.
- Verify the identity of the fetched file.

First we compile the whole hadoop project and deploy the hadoop system using docker container. Here our LRC code configuration is composed with 6 data units and 4 parity units, thus we need to deploy 10 datanodes.

Type "hadoop namenode -format" and "\$HADOOP_HOME/sbin/start-all.sh" to launch hadoop file system
Type "hdfs ec -listPolicies" to check if LRC policy is registered in hadoop.