IP'16 Apps Specs
Automatic Segmentation and Alignment of QR Codes

Description

Main Idea

Quick response (QR) code is a two-dimensional barcode that can hold different type of information. A QR code consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as a camera[1]. The QR code system became popular in many applications due to its fast readability and greater storage capacity (e.g. product tracking, item identification, document management, FCIS attendance system… to name a few).

This application aims to automatically segment the QR from an image. Then, correct its alignment so that its three corners aligned to top right, top left and bottom right, as shown in **FIGURE**.

As a start, assume that the camera is perpendicular but of different distances from the QR. The QR can be represented to the camera with any in-plane rotation angle, any backgrounds and on different materials (white paper, poster, image on mobile…etc). The application should be able to automatically segment and align the QR from the given image.

Minimum Requirements

Segment and align the QR from:

1- Images captured with perpendicular camera to the QR.
2- Different distances between camera and QR.
3- Images with different backgrounds.
4- QR with different in-plane rotation angle.
5- QR on different materials (i.e. different lighting conditions)

[1] https://en.wikipedia.org/wiki/QR_code

Possible Add-ons (Bonuses)

Segment and align the QR from:

1- Picture taken from different camera perspectives.
2- Picture of more than one QR.

Suggested Search Tracks and Keywords

You may use some/all of the following keywords as a guide (not restricted to them):

1- Segmentation.
2- Line Hough transform.
3- Morphological operations.
4- Region properties.

Test Images for Minimum Requirements

Case1: Upright shot of the QR on simple background with different distances.

Case2: Shot of the QR with any rotation angle on simple background.

Case3: Upright shot of the QR on complex background and different materials.

Case4: Shot of the QR with any rotation angle on complex background and different materials.

Test Images for Bonuses

Case5: Tilted versions of cases 1, 2, 3, 4.

Case6: Frontal shot with many QRs on different backgrounds.

Case7: Tilted versions of cases 6.

References

1- https://en.wikipedia.org/wiki/QR_code
2- Textbook Ch. 3: Intensity Transformations and Spatial Filtering
3- Textbook Ch. 9: Morphological Image Processing
4- Textbook Ch.10: Image Segmentation

Sample Input/Output

Input	Output