Image analysis methods for visual code detection and recognition

Key members: László G. Nyúl [1]
Péter Bodnár [2]
Melinda Katona [3]
Lifetime from: 2011

Short description:
The aim of this research is to study image features and processing methods for detection of visual code regions (1D barcode, 2D datamatrix, OCR characters), with special focus on highly accurate and efficient algorithms that can be adapted for real industrial applications.

Description: */
Visual codes are used in various places for various applications to identify or describe entities of our everyday life (e.g., merchandise, mail and postal packages, electronic parts). Codes can vary by type (e.g., 1D barcode, 2D datamatrix, OCR character strings), by size (e.g., a few hundred microns, a few centimeters, or even a meter), or by producing technology (e.g., printing, micrograving). Codes are often scanned by 2D digital cameras or line-scanners. The distortions of the surface that carries the code, uneven lighting and imprecise measuring all impose problems on the detection algorithms. In addition, the resources available for processing these data are also very limited and the requirements on the processing time and accuracy make the problem very challenging in real life applications.

The aim of this research is to study image features and processing methods for detection of regions containing visual codes, with special focus on highly accurate and efficient algorithms that can be adapted for real industrial applications.

We experimented with modifications of known localization methods in the state of the art, and work on novel approaches for efficient detection of visual codes.

MIN-MAX

This method treats the image as a whole, and therefore requires a fair amount of RAM and computation time. Supposed that intensity levels have been normalized, no other preprocessing operations are required since it handles noisy, blurry or distorted images well.

Knowing the maximum element size of a code pattern, we apply the morphologic gradient operator on the image with a box kernel. The next step is removing ghost elements from the feature image with a binary threshold. After that, we apply morphologic closing operation on the feature image. This is for closing gaps caused by larger blocks of elements of the same color.
Morph. gradient
Binary threshold
Opening
Contour detection
Local Clustering

Local Clustering came from examining the behaviour of textures. Most barcodes, like regular textures, can be easily identified by observing only small parts of them. These barcode parts together form the desired barcode region with known height and width. The first part of the method is partitioning the image to square tiles and look at each tile for barcode-like appearance. Each tile is assigned a value that indicates the grade of the presence of this feature. Globally, a matrix is formed from these values. Texture parts have similar local statistics in their neighbourhood, so searching this matrix for compact areas defines image ROIs representing a barcode with high possibility.
The main idea of Local clustering is that an image region that contains a barcode segment has many similar stretched pixel clusters. The minimum count of expected clusters can be derived from the widest bar of the barcode. Degree of stretch can be measured with the diameter of the cluster (defined as twice the distance of the furthest cluster point from the cluster center). With exactly horizontal or vertical lines, the largest cluster diameter is the tile size, in oblique situations, the largest cluster diameter is expected to be longer than that. Furthermore, stretched separate clusters need to be aligned approximately identically, otherwise one cluster would touch another, decreasing the number of separate clusters in a tile below our threshold. For preprocessing, we use median filter first that eliminates salt-and-pepper noise.
Contour detection
According to the previous idea, we follow the same method of partitioning. The assigned value showing barcode-like appearance is based on distance transformation of the edge map. Distance transformation is an operation that works with an initial set of points, like corners or edges. Value 0 is assigned to these points, and any others get the distance value to the closest point of the initial set. We apply Canny edge detector for this point set to the transformation. For each tile of the distance map, means and standard deviations are calculated. For 1D codes, distance values spread between half of the minimum and half of the maximum line width. For 2D codes, these values usually stay below half of their block size, but higher values are also possible, since blocks of the
same color can be next to each other in multiple directions. Having such code parts significantly raise the mean of distance values on the image tile. However, these holes in the feature matrix are rare enough so they do not split barcode-like areas.
Edge detection
Localization with Hough transformation

Hough transformation can also be used as a reliable localization approach, as barcodes consist of roughly equally long, parallel lines in a small area. Probabilistic Hough transformation gives a probabilistic estimation for detecting straight lines with the help of a subset of the edge points of the original image, outperforming the standard Hough transform. For preprocessing, we use a blur filter since smooth images are desired for the Canny edge detector.

After we obtain a list of lines with their center point, length, and orientation, we can group them to decide whether they constitute a barcode or not. We define the minimum number of lines, the proximity needed for the lines to be in the same group, and the tolerance for length and orientation from the means within the group.

In the final step, group centers are returned, and the image can be cropped for decoding with known barcode decoding implementations.
Using the ensemble of detectors increase different performance values of the single detectors based on the aggregation method. With majority voting, we can increase the precision by losing important ROIs. Using the maximum values of all feature images gives good recall, but it decreases the precision. Weighted voting, based on the separate recall values gives good recall rate while keeping precision relatively high. Since the only difference is in how we compute the values of the final feature image, those algorithms share the same running time.
In industrial setups parallel execution may also be possible for further improve detection speed. Furthermore, the data of the proposed algorithm, like the edge map can be re-used as the input for other discussed classifiers.

**Kategória:** Industrial Applications
Segmentation

**Source URL (retrieved on 2018-02-25 05:18):** [http://www.inf.u-szeged.hu/node/2216](http://www.inf.u-szeged.hu/node/2216)

**Links:**
[1] [http://www.inf.u-szeged.hu/~nyul](http://www.inf.u-szeged.hu/~nyul)
[2] [http://www.inf.u-szeged.hu/~bodnaar](http://www.inf.u-szeged.hu/~bodnaar)
[3] [http://www.inf.u-szeged.hu/~mkatona](http://www.inf.u-szeged.hu/~mkatona)