

*Nikonenko A.N., Korotka L.I.***RECOGNITION AND GENERATION OF QR/BAR CODES IN MOBILE APPLICATIONS****Ukrainian State University of Chemical Technology, Dnipro, Ukraine**

The article is devoted to the analysis of methods of recognition and generation of QR/bar codes in mobile applications. The most common algorithms for detecting and generating codes are considered: Otsu method, Reed-Solomon codes, binary coding algorithm, Orazio Gallo and Roberto Manduchi approach. In this article, it is proposed to use an algorithm based on these approaches, which is refined and used for the software implementation of the mobile application. The main stages of the recognition process in the work are: converting a color image into gray shades (using the pixel hue difference method); gray image binarization (Otsu algorithm); using affine transformations to preserve parallelism, which includes rotation, stretching, shifting, scaling. The main stages of the generating process for QR codes are: processing and reduction to the standard form for QR codes; the process of generating a binary string (in the work data of only numerical and symbolic types was considered); counting the total number of characters in the source string and converting the decimal number to a binary representation; determining the length of the generated binary string according to the Denso Wave standard (the encoding of character types is different); determining the required length of the final binary string; auxiliary tokens are concatenated; generation of Reed-Solomon error correction codes. Three types of bar codes are considered: code 128, code 39, and EAN 13. The last type was used. A bar code scanner includes two steps for automatically scanning all bar codes: finding the location of the bar code, bar code decoding. To localize the image and the bar code that is captured by the system, the algorithm of Orazio Gallo and Roberto Manduchi is used. In order to obtain a smoothed map, a block filter of a certain size was used, taking into account the size of the input code image. In addition to the code localization, it is cut off: the intensity of each pixel is taken into account, and the lines of the bar code (pixels with an intensity greater than zero) are highlighted. An image with improved contrast is converted to binary form, and to convert to an ideal image, each image column is scanned and checked for the maximum number of pixels with an intensity of zero or one. This article focuses on describing the edge detection process. The decoding algorithm uses an array of code width bands. All the above approaches and algorithms based on them are implemented as a mobile application for recognition and generation of codes.

Keywords: recognition, generation, binarization, affine transformations, bar code.

DOI: 10.32434/2521-6406-2019-5-1-36-42

Formulation of a problem

A sufficient amount of works has been devoted to recognition of texts [1], labels and barcodes. The use of QR codes (Quick Response code) in everyday life opens up new opportunities: to establish a connection between hyperspace and real life, using modern technologies for transmitting digital data streams and methods of indirect information transfer. QR codes are often used for commercial tracking, travel tickets, storage of personal information, etc.

As is well known, bar codes are read with scanners that use laser beams. Currently, smartphones can also be used as scanning devices for QR codes. This is a very simple, easy and convenient way to access specific URLs or websites. The fact that there is no need to manually enter the URL provides a better conversion, and therefore it is good for marketing purposes and more.

Quite contradictory requirements are put forward for QR codes: the code must store a huge

amount of necessary information, however, at the same time occupy a small area. In addition, if the information is damaged, then it should be easily restored. In this case, the code must have a high recognition rate, despite the fact that some of it may be missing. Obviously, it is quite convenient and practical to scan information using mobile phones where a recognition program is installed. Obviously, the development of free mobile applications for recognition and generation of codes is relevant.

Analysis of recent research and publications

It should be noted that, despite the practical application of codes and the relevance of their use, there are not so many works on their detection and generation. As you know, the concept of a QR code first appeared in Japan in 1994: a code standard was developed and presented to track the different stages of Toyota production at its enterprises. Due to simplicity, the popularity of QR codes has increased, which made it possible to use these two-dimensional codes in all spheres of human life [2]. The most common is the algorithm proposed by Otsu, calculating the binarization threshold for a halftone image, which is used in the field of computer pattern recognition and image processing to produce black and white images [3].

One approach to decoding bar codes was proposed by Orazio Gallo and Roberto Manduchi. The method is based on deformable patterns and uses all the information about the gray level of each pixel. They presented a new bar code decoding algorithm (localization and reading) that can work with blurred, noisy images and low-resolution images [4].

In the general case, when generating noise-proof codes (Reed-Solomon), multiplication and division of polynomials into each other is used [5]. The algorithm proposed by Luping Fang and Chao Xie can detect a bar code area even against a complex background using image analysis based on areas. Unlike traditional image analysis based on regions, the algorithm is processed using an integral image. It can work with images that are blurred, contain obliquely arranged barcode areas and are shot under different lighting conditions. The fast and reliable way to detect a noisy barcode is offered by Xianyong Fang [6–8].

Formulation of the objectives of the article (problem statement)

The paper discusses the methods of recognition and generation of QR / bar codes that are available in the public domain and analyzes them. An aim of the work is to develop an algorithm for generating and detecting codes, on the basis of which a mobile

application is created. In this case, special attention is paid to the performance of the developed algorithm.

Presentation of the main research material

As is known, the use of two-dimensional bar codes is that the printed code is transmitted to the object, then read using a mobile device that has a functioning camera and installed necessary software that will decode the bar code itself.

The QR code detection algorithm presented in this work consists of several stages: searching for outlines in the image; contour approximations; search for suitable quadrangles; FP block ratio checker (Finder Pattern).

There are not many programs that read and decode QR codes, they are all implemented using a simple code detection algorithm on the image obtained from the camera, after which a simple procedure for information decoding takes place. And all this requires a completely clear positioning of the selected area on the device and a special arrangement of QR codes, the area must exactly coincide with sides of QR codes. That is, the user needs to independently focus on a necessary QR code and control that the area for QR code scanning coincides with the decoded QR code itself.

Let us dwell on the important points of this process. First of all, you need to convert a color image into an image with gray shades. A method based on the difference in the illumination of pixels is used. It is a modifiable analogue of the method of averaging values of the three components of each pixel [2]. Calculated by the formula: $P[i][j] = (R + G + B) / 3$.

For this work, the selected coefficients of color components are: $P_{\text{new}} = 0.21 \cdot R + 0.71 \cdot G + 0.07 \cdot B$.

The result will be the calculation of all pixels in the image by the formula, after which a new image only in shades of gray is obtained.

For binarization of the image, the Otsu method is used, where the image forms a bimodal histogram. Next, the optimal threshold is computed, which divides the two classes so that their intragroup dispersion tends to the minimum value. The method is described in a way below [3].

Let q_1 and q_2 be the probabilities of each class, which were calculated using the formulas:

$$q_1(t) = \sum_{i=1}^t P(x);$$

$$q_2(t) = \sum_{i=t+1}^I P(x).$$

Dispersion for each of the classes is determined as follows:

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)};$$

$$\sigma_2^2(t) = \sum_{i=t+1}^l [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}.$$

The mean value of each class is calculated:

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)};$$

$$\mu_2(t) = \sum_{i=t+1}^l \frac{iP(i)}{q_2(t)}.$$

In order to get a result you need to maximize the following equation:

$$\begin{aligned} \sigma_b^2 &= \sigma^2 - \sigma_w^2(t) = \\ &= q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2. \end{aligned}$$

As a result, the required threshold was obtained, by which it is possible to divide the histogram into two classes and to assign the pixels of the smaller class to zeros, and to the pixels of the other class to ones. Thus, the binarized image is obtained and proceed to the next stage.

As is known, affine transformations retain parallelism. The full group of affine transformations consists of four transformations:

- turn;
- stretching;
- shift;
- scaling.

Transformations of this group can be written in different ways, for example, [4].

The process of coding QR codes is divided into several main stages. The process, which is implemented in this work, is described below.

The first stage is that a string with input data is processed and reduced to the standard form for QR codes.

After the string with the initial data is received, the process of generating a binary string begins. Four technical bits are added to the beginning of the binary string X, which are determined based on the data type being encoded. The list of possible combinations

includes: 0001 – numeric data, 0010 – alphanumeric data, 0100 – binary information, 1000 – special Japanese characters. In order to simplify the process of displaying the results, only numerical and symbolic data were considered in the paper.

In the next generation step, the total number of characters in the source string is calculated and the decimal number is converted to a binary representation. It is necessary to determine a length of the generated binary string according to the Denso Wave standard.

In order to encode a character data type, the source string must be broken into pairs of characters. Then you need to know the value from the ASCII table for the first character in each of the pairs, this value is multiplied by the number 45, then the value is taken according to the same ASCII table for the second character in each of the pairs. These two values are summed, and each pair of symbols has its own numerical value. The resulting numbers must be sequentially represented as 11-bit numbers in binary form. If the line contains an odd number of characters, then the value for this character from ASCII is taken and converted to a 6-bit binary number. All received blocks are also concatenated with the generated string X on the right side.

After completing the described steps, it is necessary to refer to the version table of Denso Wave again, and to determine the required length of the final binary string. If the length of the string X is less than required, then from the right side from 1 to 4 bits are added (depending on how many characters are missing). The generated string X is divided into 8-bit tokens (blocks) and, if the last block is less than 8 bits, then zeros are appended to it on the right side.

At the final stage of generating a binary string, auxiliary tokens are concatenated with string X. In case the generated string is smaller than required (according to the version table), eight bit blocks are added to the end of the generated string in turn, until the required number of bits is reached: 11101100 and 00010001. When the generated string contains the required number of bits, the binary string is considered to be finally generated and can be used in the next steps of generating QR codes.

Next, go to the generation of Reed-Solomon codes for error correction. As noted earlier, in general case, when generating Reed-Solomon codes, the multiplication and division of polynomials are used. For example, if polynomial A is a code word (generated on the basis of the binary string obtained in the previous step), and B is an irreducible polynomial that is known to both sides exchanging

information, then their multiplication will generate a new polynomial C.

This process means that when the transmitted message is received, the process of decoding it will be to divide the polynomials by each other. If the division of a polynomial C by a polynomial B leaves a remainder, then a transmitted message contains an error. The remainder of the division forms a polynomial syndrome, through which you can find out the damaged bits. In addition, in some cases, you can correct a certain number of errors in the received message. If the degree of polynomial B differs from the degree of polynomial A by at least two, then it is possible not only to determine the presence or absence of an error in the transmitted message, but also to correct some of them. In this case, the following equation (where k is the degree of the polynomial; t is the maximum value of possible corrections) shows that the redundancy of Reed-Solomon codes is: $k=2 \cdot t$.

The process of generating Reed-Solomon codes, their encoding and decoding, is described in the appendix of the QR codes specification [5]. As a result of generating a string using Reed-Solomon codes for the original message, a set of decimal values will be generated. They need to be converted to binary format and concatenated with the source line on the right side. The result of all the previous steps will be a string in binary format containing the original message data plus Reed-Solomon codes in order to provide the possibility of error correction in the decoded message. Using Reed-Solomon codes, code words are generated in order to recover errors when reading information from a QR code. Then they are combined with the original message, which was generated at the first stage of the algorithm. The result will be one line containing a certain number of bits according to the specification. At this stage, the QR code has not yet been formed. Next, we analyze how the two-dimensional representation of a QR code is generated. To begin with, a standard QR code pattern is generated, which contain all the necessary technical bits needed to define auxiliary information. Such standard template is the basis of any QR code. This template always includes three positioning markers located in an upper left and right corner, as well as in a lower left corner of a template. In addition, all the templates contain special Timing Patterns (alternating white and black pixels), which serve to pre-determine the version of the QR code.

With a known version of the QR code and a level of error correction, and in the presence of all the technical primitives in the template, the filling of the template of the QR code with previously

generated information begins. The process of adding bits of the original message to the matrix is as follows: first, the first pixel (a color is selected according to the value of the bit) is located in the lower right-most corner of the template. The second pixel is added to the left of the first on the same line. The next pair of bits is located on top, similar to the first two. This is how the entire template column is filled. When the filled column has reached the top of the template, the process begins anew, but from the first free cell located on the top-right (the same process, but only from the bottom upwards), as well as the first column. Thus, the entire QR code template is filled. The received QR code is not fully completed at this stage. In order to complete the code generation process, it is necessary to apply special masks and calculate penalty values from the received QR codes. According to the specification, it is necessary to generate eight different types of QR codes based on the newly generated one. Each of these eight codes is defined by its formula, presented in the table.

Each of the formulas presented is used to determine whether a pixel located with certain indices should be replaced with the opposite value.

A bar code is a visual display of information in the form of stripes and gaps on the surface. Stripes and spaces can be of different widths and consist of numbers, symbols. Today, different types of bar codes are used (code 128, code 39 and EAN 13) [6–8]. In this case, EAN 13 is used.

Formulas for generating eight codes

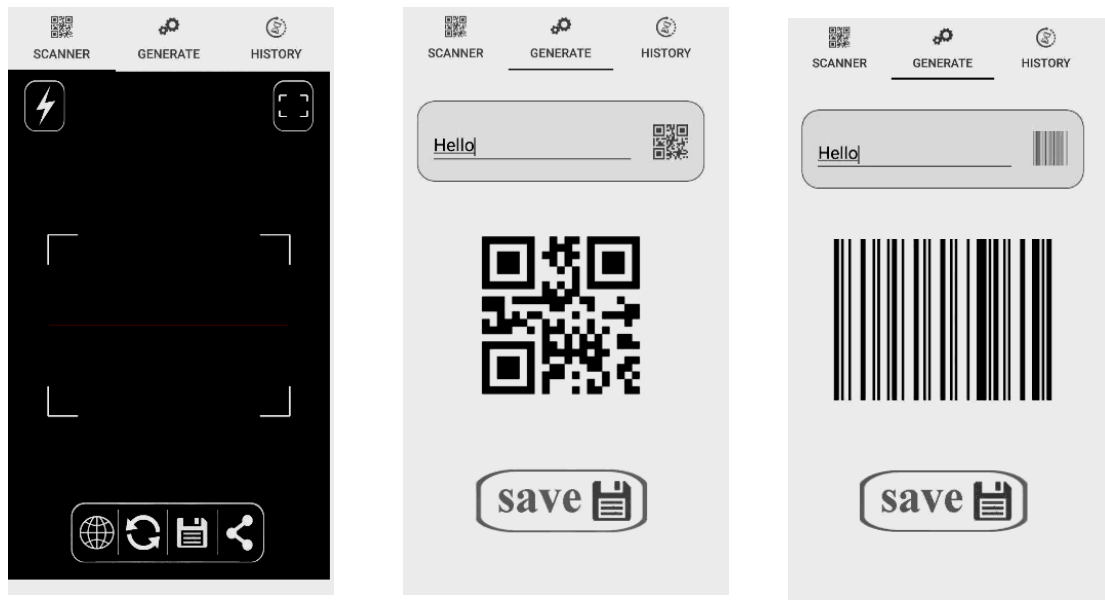
Mask Pattern	Condition
0	$(i+j) \bmod 2=0$
1	$i \bmod 2=0$
2	$j \bmod 3=0$
3	$(i+j) \bmod 3=0$
4	$((i \div 2)+(j \div 3)) \bmod 2=0$
5	$(i \cdot j) \bmod 2+(i \cdot j) \bmod 3=0$
6	$((i \cdot j) \bmod 2+(i \cdot j) \bmod 3) \bmod 2=0$
7	$((i \cdot j) \bmod 3+(i+j) \bmod 2) \bmod 2=0$

Here i is the row index, j is the column index.

The BAR code scanner includes several bar code scanning steps:

- finding the location of the bar code;
- bar code decoding.

The numbers are divided into three groups, that is, they form three different sets of numbers. The first group is encoded using a scheme, where each digit has two possible codings: one with even parity and the other with odd parity. The first digit is



Windows of the mobile application «Recognition and generation of QR/bar codes»

encoded with the choice of pattern. Each digit in the last group of six digits is encoded using one set of column patterns that are used for the Universal Product Code (UPC). When the first digit is zero, then the digits in the first group of six are encoded using patterns that are used for the UPC; because of this, the UPC bar code is treated as an EAN-13 bar code with the first digit shown as zero [6].

In this paper, the algorithm by Roberto Manduchi and Orazio Gallo was chosen, which states that the bar code image is captured by the camera in such way that the vertical axis is parallel to the stripes. Also, this image is transformed into an image in a gray tint [7]. It is necessary to calculate the horizontal and vertical derivatives $I_x(n)$ and $I_y(n)$ for each pixel n , then they are combined in a non-linear way, which is calculated by the formula: $I_e(n) = |I_x(n)| - |I_y(n)|$.

It is possible to suggest that the majority of points in the bar code must have a large $I_e(n)$ value. To achieve it, a block filter of size 31×31 is used over $I_e(n)$, and a smoothed map $I_s(n)$ is obtained.

The filter size is selected from the size of the incoming bar code image and the smallest bar code size. As a result, the binary form $I_s(n)$ is presented with a threshold, which is calculated using the simple Otsu method, and the output image. Next, the pixel N_0 is selected, which tends to the maximum $I_s(n)$. The vertical and horizontal line extends from N_0 and a rectangle is formulated, which has sides parallel to the axes of the bar code image.

As is well known, when a bar code is scanned, unnecessary information that surrounds it enters the camera. Therefore, when the process of localization

of the bar code passes, the information surrounding it will be deleted (the pixels are zero). The image is then cropped: the lines of the bar code are highlighted (the pixels are greater than zero).

Since an image can be distorted in real time, it needs to be converted to perfect. To do this, you need to check each column for the maximum number of pixels with an intensity of zero or one. If a column has more pixels with one, then the column needs to be converted to pixels with intensity of one. In order to detect edges, you need to create an array of edges that forms the edges of the bar code image strips. The bandwidth is calculated by subtracting the successive elements of the array of edges (the bandwidth is the input to the decoding).

Decoding occurs in such a way that the bar code number is decoded using an array of width bands. The first digit does not change (because the first digit is the country code, it will not change). The first group, which includes six digits, is encoded as LGLGGL, and the remaining group, which also consists of six digits, is encoded as RRRRRR (the black and white columns alternate where white is zero and black is one) [7].

All of the above is implemented as a mobile application, some of the results of its use are shown in the figure.

Results and conclusions

The article describes an approach to developing a mobile application for recognizing and decoding QR/bar codes.

As a result, the following tasks were solved:

- algorithms for the transformation of

geometric objects on the plane, coding and decoding of QR/ bar codes, recognition and image processing were studied;

- algorithms for the selection of QR/bar code in the image;

- algorithms necessary for converting QR codes to standard form using affine transformations are presented;

- implemented QR/bar code decoding algorithm;

- developed a mobile application based on the described algorithm.

REFERENCES

1. Korotka L.I., Nikonenko A.N. Mathematical methods for processing multidimensional data // MATERIALS III-rd International scientific-technical conference COMPUTER MODELING AND OPTIMIZATION OF COMPLEX SYSTEMS (Dnipro, 1-3 November 2017). – Dnipro: USUCT, 2017. – P.116-117.
2. ISO/IEC 18004:2015 Information – Automatic identification and data capture techniques – QR Code BARcode symbology specification. Available at: <https://www.iso.org/standard/62021.html>
3. ISO/IEC 18004:2000 Information technology – Automatic identification and data capture techniques – BAR code symbology [Электронный ресурс] – QR code. Available at: <https://www.iso.org/standard/30789.html>
4. Otsu N.A Threshold Selection Method from Gray-Level Histograms // IEEE Trans. Sys. Man., Cyber. – 1979. – Vol.9. – Issue 1. – P.62-66.
5. David F. Rogers, J. Alan Adams. Mathematical elements for computer graphics // Second Edition. McGraw Hill Education. – 2012. – 611 p.
6. University. EE387 Notes #7, 2014. – [Электронный ресурс] P.24-86. Available at: <https://web.archive.org/web/20140630172526/http://web.stanford.edu/class/ee387/handouts/notes7>.
7. Chai D, Hock F. “Locating and Decoding EAN-13 BARcodes from Images Captured by Digital Cameras”, Information, Communications and Signal Processing. – Fifth International Conference. – 2005. – P.1595-1599.
8. Gallo O., Manduchi R. “Reading 1D Bar-codes with Mobile Phones Using Deformable Templates”, Pattern Analysis and Machine Intelligence, IEEE Transactions. – 2011. – Vol.33. – № 9. – P.1834-1843.

Received 08.05.2019

РОЗПІЗНАВАННЯ ТА ГЕНЕРАЦІЯ QR/BAR-КОДІВ У МОБІЛЬНИХ ДОДАТКАХ

Никоненко О.М., Коротка Л.І.

Стаття присвячена аналізу методів розпізнавання та генерації QR/bar кодів в мобільних додатках. Розглянуто найбільш поширені алгоритми детектування і кодування кодів: метод Оцу, коди Ріда-Соломона, алгоритм двійкового кодування, підхід Ораціо Галло і Роберто Мандучі. У роботі пропонується використовувати алгоритм, який базується на зазначених підходах, та, який доопрацьовано і використано для програмної реалізації мобільного додатку. Основними етапами процесу розпізнавання в роботі є: конвертація кольорового зображення у відтінки сірого (використовується метод різниці відтінків пікселів); бінаризація сірого зображення (алгоритм Оцу); використання афінних перетворень для збереження паралельності, які включають поворот, розтягнення, зрушення, масштабування. Основними етапами процесу генерації QR-кодів є: обробка і приведення до стандартного для QR кодів виду; процес генерації бінарних рядків (в роботі розглядалися дані тільки чисельного і символічного типів); підрахунок загальної кількості символів у заданому рядку і перетворення десятичного числа в бінарне надання; визначення довжини бінарного рядка, що генерується, відповідно до стандарту компанії Denso Wave (кодування символічних типів здійснюється інакше); визначення необхідної довжини кінцевого бінарного рядка; доповніжні токени конкатенуються; генерація кодів Ріда-Соломона для корекції помилок. В роботі розглянуті три типи штрих-кодів: код 128, код 39 і EAN 13. Використано останній тип. Сканер штрих-коду включає в себе два кроки для автоматичного сканування всіх штрих-кодів: знаходження розташування штрих-коду; декодування штрих-коду. Для локалізації зображення штрих-коду, який захоплюється системою, використаний алгоритм Ораціо Галло і Роберто Мандучі. З метою отримання згладженої карти, використано блокувальний фільтр певного розміру з урахуванням розміру вхідного зображення коду. Крім локалізації коду, виконується його обрізка: розглядається інтенсивність кожного пікселя і виділяються рядки штрих-коду (пікселі з інтенсивністю більше нуля). Зображення з поліпшеною контрастністю перетворюється в двійкову форму для перетворення в ідеальне зображення, кожен стовпець зображення сканується і перевіряється на максимальну кількість пікселів з інтенсивністю нуль або одиниця. В роботі окрему увагу приділено опису процесу виявлення границь зображення. Алгоритм декодування використовує масив смуг ширини коду. Всі вище викладені підходи та на їх основі алгоритми, реалізовані у вигляді мобільного додатку розпізнавання та генерації кодів.

Ключові слова: розпізнавання, генерація, бінаризація, афінне перетворення, штрих-код.

РАСПОЗНАВАНИЕ И ГЕНЕРАЦИЯ QR/BAR-КОДОВ В МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ

Никоненко А.Н., Короткая Л.И.

Статья посвящена анализу методов распознавания и генерации QR/bar кодов в мобильных приложениях. Рассмотрены наиболее распространенные алгоритмы детектирования и кодирования кодов: метод Оцу, коды Рид-Соломона, алгоритм двоичного кодирования, подход Орацио Галло и Роберто Мандучи. В работе предлагается использовать алгоритм, базирующийся на указанных подходах, который доработан и использован для программной реализации мобильного приложения. Основными этапами процесса распознавания в работе являются: конвертация цветного изображения в оттенки серого (используется метод разности оттенков пикселей); бинариза-

ция серого изображения (алгоритм Оцу); использование аффинных преобразований для сохранения параллельности, которые включают поворот, растяжение, сдвиг, масштабирование. Основными этапами процесса генерации QR-кодов являются: обработка и приведение к стандартному для QR кодов виду; процесс генерации бинарной строки (в работе рассматривались данные только численного и символического типов); подсчет общего количества символов в исходной строке и преобразование десятичного числа в бинарное представление; определение длины генерируемой бинарной строки согласно стандарту компании Denso Wave (кодирование символьных типов осуществляется иначе); определение необходимой длины конечной бинарной строки; вспомогательные токены конкатенируются; генерация кодов Рида-Соломона для коррекции ошибок. В работе рассмотрены три типа штрих-кодов: код 128, код 39 и EAN 13. Использован последний тип. Сканер штрих-кода включает в себя два шага для автоматического сканирования всех штрих-кодов: нахождение местоположения штрих-кода; декодирование штрих-кода. Для локализации изображения штрих-кода, который захватывается системой, использован алгоритм Орацио Галло и Роберто Мандучи. С целью получения сглаженной карты, использован блочный фильтр определенного размера с учетом размера входного изображения кода. Помимо локализации кода, выполняется его обрезка: рассматривается интенсивность каждого пикселя и выделяются строки штрих-кода (пиксели с интенсивностью больше нуля). Изображение с улучшенной контрастностью преобразуется в двоичную форму и для преобразования в идеальное изображение, каждый столбец изображения сканируется и проверяется на максимальное количество пикселей с интенсивностью ноль или единица. В работе отдельное внимание уделено описанию процесса обнаружения краев. Алгоритм декодирования использует массив полос ширины кода. Все выше изложенные подходы и на их основе алгоритмы, реализованы в виде мобильного приложения распознавания и генерации кодов.

Ключевые слова: распознавание, генерация, бинаризация, аффинные преобразования, штрих-код.

RECOGNITION AND GENERATION OF QR/BAR CODES IN MOBILE APPLICATIONS

Nikonenko A.N., Korotka L.I.

Ukrainian State University of Chemical Technology, Dnipro, Ukraine

The article is devoted to the analysis of methods of recognition and generation of QR/bar codes in mobile applications. The most common algorithms for detecting and generating codes are considered: Otsu method, Reed-Solomon codes, binary coding algorithm, Orazio Gallo and Roberto Manduchi approach. In this article, it is proposed to use an algorithm based on these approaches, which is refined and used for the software implementation of the mobile application. The main stages of the recognition process in the work are: converting a color image into gray shades (using the pixel hue difference method); gray image binarization (Otsu algorithm); using affine transformations to preserve parallelism, which includes rotation, stretching, shifting, scaling. The main stages of the generating process for QR codes are: processing and reduction to the standard form for QR codes; the process of generating a binary string (in the work data of only numerical and symbolic types was considered); counting the total number of characters in the source string and converting the decimal number to a binary representation; determining the length of the generated binary string according to the Denso Wave standard (the encoding of character types is different); determining the required length of the final binary string; auxiliary tokens are concatenated; generation of Reed-Solomon error correction codes. Three types of bar codes are considered: code 128, code 39, and EAN 13. The last type was used. A bar code scanner includes two steps for automatically

scanning all bar codes: finding the location of the bar code, bar code decoding. To localize the image and the bar code that is captured by the system, the algorithm of Orazio Gallo and Roberto Manduchi is used. In order to obtain a smoothed map, a block filter of a certain size was used, taking into account the size of the input code image. In addition to the code localization, it is cut off: the intensity of each pixel is taken into account, and the lines of the bar code (pixels with an intensity greater than zero) are highlighted. An image with improved contrast is converted to binary form, and to convert to an ideal image, each image column is scanned and checked for the maximum number of pixels with an intensity of zero or one. This article focuses on describing the edge detection process. The decoding algorithm uses an array of code width bands. All the above approaches and algorithms based on them are implemented as a mobile application for recognition and generation of codes.

Keywords: recognition, generation, binarization, affine transformations, bar code.

REFERENCES

1. Korotka L.I., Nikonenko A.N. Mathematical methods for processing multidimensional data // MATERIALS III-rd International scientific-technical conference COMPUTER MODELING AND OPTIMIZATION OF COMPLEX SYSTEMS (Dnipro, 1-3 November 2017). Dnipro: USUCT, 2017, pp. 116-117 (DOI: <http://dx.doi.org/10.32434/CMOCS-2017>).
2. ISO/IEC 18004:2015 Information – Automatic identification and data capture techniques – QR Code BARcode symbology specification. Available at: <https://www.iso.org/standard/62021.html>
3. ISO/IEC 18004:2000 Information technology – Automatic identification and data capture techniques – BAR code symbology – QR code. Available at: <https://www.iso.org/standard/30789.html>
4. Otsu N. A Threshold Selection Method from Gray-Level Histograms // IEEE Trans. Sys., Man., Cyber. 1979, vol.9, Issue: 1, pp. 62-66. (DOI: 10.1109/TSMC.1979.4310076)
5. David F. Rogers, J. Alan Adams Mathematical elements for computer graphics // Second Edition. McGraw Hill Education, 2012, 611 p.
6. Gill, John Definition of BCH codes // Stanford University. EE387 Notes #7. 2014, pp. 24-86. Available at: <https://web.archive.org/web/20140630172526/http://web.stanford.edu/class/ee387/handouts/notes7>.
7. Chai D, Hock F, "Locating and Decoding EAN-13 BARcodes from Images Captured by Digital Cameras," Information, Communications and Signal Processing, Fifth International Conference, 2005, pp. 1595-1599, doi: 10.1109/ICICS.2005.1689328
8. Gallo, O., Manduchi, R., "Reading 1D BARcodes with Mobile Phones Using Deformable Templates," Pattern Analysis and Machine Intelligence, IEEE Transactions, 2011, vol.33, no.9, pp.1834-1843 (doi: 10.1109/TPAMI.2010.229).