Management Systems
in
Production Engineering

2020, Volume 28, Issue 3, pp. 154-161

Date of submission of the article to the Editor: 02/2020
Date of acceptance of the article by the Editor: 06/2020

DOI 10.2478/mspe-2020-0023

RECOGNITION OF DATA MATRIX CODES IN IMAGES AND THEIR APPLICATIONS IN PRODUCTION PROCESSES

Ladislav KARRACH, Elena PIVARČIOVÁ
Technical University in Zvolen

Abstract:
Data Matrix codes can be a significant factor in increasing productivity and efficiency in production processes. An important point in deploying Data Matrix codes is their recognition and decoding. In this paper is presented a computationally efficient algorithm for locating Data Matrix codes in the images. Image areas that may contain the Data Matrix code are to be identified firstly. To identify these areas, the thresholding, connected components labelling and examining outer bounding-box of the continuous regions is used. Subsequently, to determine the boundaries of the Data Matrix code more precisely, we work with the difference of adjacent projections around the Finder Pattern. The dimensions of the Data Matrix code are determined by analyzing the local extremes around the Timing Pattern. We verified the proposed method on a testing set of synthetic and real scene images and compared it with the results of other open-source and commercial solutions. The proposed method has achieved better results than competitive commercial solutions.

Key words: *adaptive thresholding, connected component labelling, data matrix code, finder pattern, timing pattern*

INTRODUCTION

Every company that wants to maintain its competitiveness in a demanding market must permanently ensure a continuous increase in efficiency and labour productivity, and that can be done only by continuous improvement of its internal processes. Only in this way can it ensure that the prices of its products do not rise more than the market is ready to accept. In doing so, the company must also guarantee an increase in the customer satisfaction. Here is offered e.g. reorganization of business processes, introduction of modern technologies, and use of cheaper or newer materials. Another possibility in streamlining processes in the company is to take advantage of automatic identification. In a company processes, it is necessary to identify and track a number of elements and components in its operation. A man can identify objects by the senses and classify them into categories, and work with them. However, he can also use modern technologies and means of automatic identification. Modern automatic identification technologies can work faster, more precisely and more efficiently than humans. They can reduce process costs, speed up and streamline production processes, ensuring productivity growth and gaining competitive edge.

The history of automatic identification is long, but barcodes were a major breakthrough in the development of automatic identification. They were first patented in 1952, however it took over twenty years before they became commercially successful. Traditional barcodes are one-dimensional and their size and capacity is often no longer sufficient to store the necessary data. Therefore, they are gradually replaced by large-capacity two-dimensional matrix codes, which allow storing considerably more information.

In 1987, Intermec Corporation introduced the first high-capacity stacked barcode named Code 49.

In 1991, Symbol Technologies introduced a 2D code named PDF 417. In addition to common text, also graphics and programming instructions could be encoded.

In 1994, Denso Wave developed a QR code. The code has begun to be used in the automotive industry (for tracking parts during vehicle assembly), but has quickly spread to other areas.

The Data Matrix code was invented in 1994 by International Data Matrix, Inc. (ID Matrix).

2D matrix codes are today widely used in areas like industrial automation (labelling of materials, parts and products), intelligent logistics centres (package labelling), package tracking, product life cycle management and quality control (automated tracking of products), warehouse management, retail sale (scanning items at cash register), healthcare (patient identification including they medical history), libraries (book identification), e-payments (payment instructions), e-cards (contact information), advertising, marketing, tourism (links to websites which contains more information about object on which 2D code is placed) etc. The rapid development and massive use of mobile phones has still increased the significance of 2D codes.

The Data Matrix code

The Data Matrix code contains several characteristic components (Fig. 1), which allow to distinguish a Data Matrix code among other objects in an image, and also to distinguish it from other two-dimensional matrix barcodes. The Data Matrix code is composed of black (dark) and white (light) squares called modules (or cells). One module represents one bit: usually black for the "1" and white for the "0". These modules are organized into rows and column and form squared (mostly used) or rectangular Data Matrix code. Data Matrix code is structured as follows:

- Finder Pattern – consists of the two perpendicular solid lines (consisting of black modules only) on the outside of the Data Matrix code that form an "L" shape. Finder Pattern is used by readers to determine position and orientation of a Data Matrix code.
- Timing Pattern – is placed on the two opposite sides to Finder Pattern, where alternate black and white modules. Timing Pattern (Clock Pattern) is used to determine the size of a module, the number of rows and columns, and possible distortion of code.
- Quiet Zone – is a white area of width at least one module located around the Data Matrix code. Quiet Zone should not contain any patterns or structures which can confuse readers.
- Data – are encoded inside Data Matrix code and are protected by an error correction carried out via a Reed-Solomon algorithm (allow restoration of damaged data). This also means that a Data Matrix code can be partially damaged up to approx. 25% and can still be entirely read out.

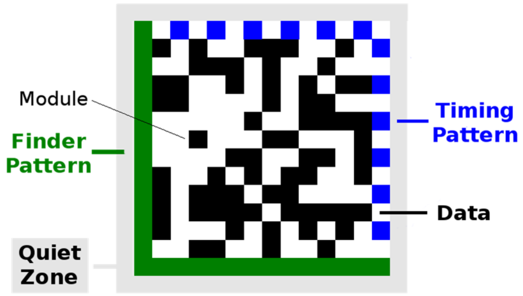


Fig. 1 Structure of a Data Matrix code

The size of Data Matrix code depends on the type of data (numeric, character, binary) and the length of the data which should be encoded. For Data Matrix codes (ECC200) the size can vary from 10×10 to 144×144 modules (only even number of modules) and maximal data capacity is 3116 numbers, 2335 alphanumeric characters, or 1556 bytes. If a size of a code is more than 26×26 modules, it is divided in blocks (Fig. 2). Since the minimal size of Data Matrix is the smallest one among various barcode types, it is especially suitable for marking small items.

Data Matrix codes are standardized by an international organization (ISO) and it can be used free of any licensing or royalties.

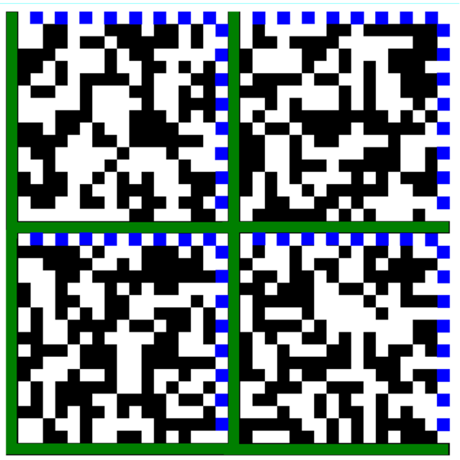


Fig. 2 Structure of a bigger Data Matrix code with 4 blocks

To read out Data Matrix codes, camera based device like smartphone, mobile computer or hand scanner is needed.

Data Matrix code and Direct Part Marking

Direct Part Marking (DPM) is a method how to permanently attach information to the part. DPM allows the Data Matrix code to be directly applied to a part (without need to print and place a label on it) using different marking methods. Data Matrix codes can be marked directly onto components via laser marking, embossing, engraving, ink-jet, electro-chemical etching, etc. In most instances, the permanent mark applied will last for the lifetime of the component and in contrast to label could not fall off or be removed.

In Fig. 3a there are Data Matrix codes that are printed on product packaging, or on labels sticked to products in contrast in Fig. 3b, where are Data Matrix codes that are laser marked directly onto the surface of products or parts (DPM).

Fig. 3 Examples of usage and application of Data Matrix codes

Data Matrix code and RFID

When goes to automation and efficiency, the Data Matrix codes can be printed on labels (Fig. 4) which contain also RFID tags (transponders). RFID (Radio Frequency IDentification) tags can store any kind of information and/or data related to product. The RFID technology provides some advantages over stand-alone Data Matrix codes:

- the tags can be read without direct visibility to the reader and over longer distances (tags can be placed on product inside package),
- several tags can be read simultaneously,
- tags can be overwritten multiple times.

Fig. 4 Label with printed Data Matrix code and RFID chip

On the other hand, Data Matrix codes are usually simpler, more practical, and cost-effective compared to RFID solutions.

In Fig. 4 there is a label with printed Data Matrix code and part of the label is also RFID chip with the antenna. Printed Data Matrix code can be scanned optically (if it is visible to scanning device), and at the same time it is possible to read information encoded in the RFID tag (if visibility is not guaranteed). On the market there are printers that allow print the label and at the same time write information into the RFID tag.

METHODOLOGY OF RESEARCH

For recognizing the Data Matrix codes, which are part of an image in industrial world, we have designed and tested a computationally efficient method for locating the Data Matrix codes in the images. The proposed Data Matrix code localization method uses typical Finder Pattern and Timing Pattern patterns, local thresholding, connection of adjoining points into continuous regions, and external region boundaries to identify Data Matrix candidates in image. This method follows the work published in [13].

The proposed algorithm for location of the Data Matrix code in the image is described by following steps (to illustrate individual steps, the image of the Data Matrix code marked with laser on aluminium cast, is used Fig. 5).

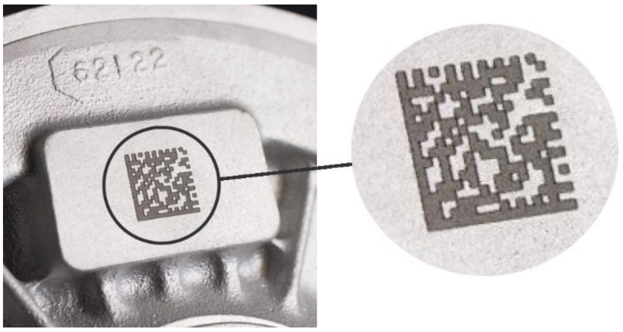


Fig. 5 Data Matrix code engraved by laser on aluminium casting
Source: [19].

1. Colour conversion from RGB to gray scale image (Fig. 6)
Input image acquired by camera device in RGB colour model is converted to gray scale image using equation:

$$I = \frac{77R + 151G + 28B}{256} \tag{1}$$

where:
I stands for gray level and *R*, *G*, *B* for red, green, and blue colour intensities in the RGB model.

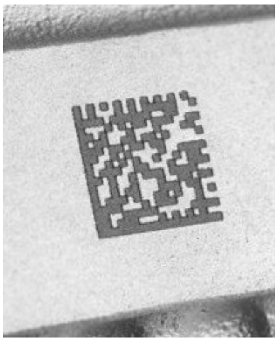


Fig. 6 Grayscale image

2. Binarization
Next, the gray scale image is converted into a binary image (Fig. 7) using modified adaptive thresholding technique (Eq. 2) [4, 12, 14]. Dark points, which belong to Data Matrix code, will become foreground black points. Local threshold value *T* is computed as:

$$T(x,y) = m(x,y) - \frac{I(x,y)}{k_1} - \frac{s^2(x,y)}{k_2} \tag{2}$$

where:
$m(x, y)$ is a local mean,
$s^2(x, y)$ is a local variance of the pixels under the sliding window at coordinates (x, y),
k_1 is a constant that is controlling penalization of bright points,
k_2 is a constant that is controlling decreasing of local threshold for points in which neighbourhood intensity significantly varies (k_1 and k_2 were empirically set to 10 and 120 respectively and size of sliding window was 35).

To speed-up the threshold calculation, we have used precalculated integral sum image [4].

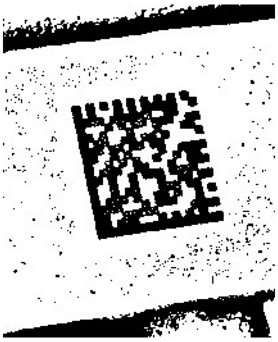


Fig. 7 Binary image

3. Connected Component Labelling

All black points in binarized image that are connected to each other are joined to continuous regions (BLOBs) [18]. These distinct regions are coloured by different colours in Fig. 8. As points are joined into distinct regions, region descriptor for each region is maintained. After all black points are connected and regions are formed, each region is described by:

- outer bounding-box (octagon defined by 8 boundary points: Top-Left, Top-Right, Right-Top, Right-Bottom, Bottom-Right, Bottom-Left, Left-Bottom, and Left-Top);
- area (the number of black points that make up the region).

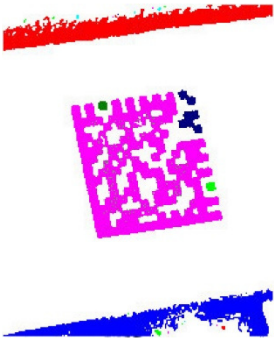


Fig. 8 BLOBs

4. Filter out unsuitable regions

Filter out regions which do not meet minimal area and aspect ratio conditions. Ignore small regions, which cannot set-up the Finder Pattern, where area < 80 and non-square regions, where aspect ratio is not in the range (0.5-2.0). In Fig. 9 only one region, coloured in magenta, meet these conditions.

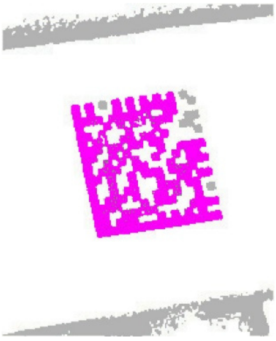


Fig. 9 Region of interest

5. Check Finder Pattern

Check if it is possible to select from 8 boundary points describing region (Fig. 10a), such 3 points that would form vertices of right-angled isosceles triangle.

Take 3 subsequent points P_A-P_B-P_C from groups of border points P1-P3-P5-P7 and P2-P4-P6-P8 and check if they satisfy:

$|P_A, P_C|^2 \cong |P_A, P_B|^2 + |P_B, P_C|^2$ and $|P_A, P_B| \cong |P_B, P_C|$ (Fig. 10b).

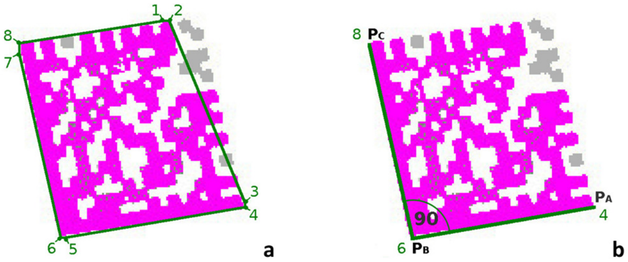


Fig. 10a,b Boundary points and three boundary points that form right-angled triangle

6. Triangle leg location refinement

If right-angled isosceles triangle is identified (Fig. 10b), legs defined by line segments P_A-P_B and P_B-P_C may not be ideally aligned on region boundary (Fig. 11).

For searching for the more exact location of boundary points of Finder Pattern projection along the estimated line segments and searching for highest difference of adjacent projections is used.

Fig. 11 Initial position of the line segment P_A-P_B

7. Optimizing location of line segments P_A-P_B and P_B-P_C

7.1. Calculate projection (sum of the points belonging to region *O*) in rectangular area of width 7 points along the line segment P_A-P_B (Fig. 12). In this projection, look for maximal difference between two adjacent beams.

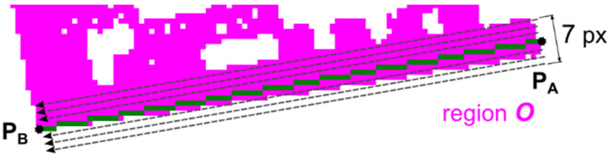


Fig. 12 Projection along the line segment P_A-P_B

7.2. Subsequently shift the points P_A, P_B ±4 points perpendicular to the initial position of line segment P_A-P_B, and for every move repeat the paragraph 7.1 (Fig. 13).

***Fig. 13* Shifts of the endpoints of the line segment P_A-P_B**

7.3. More exact position of points P_A, P_B is such where the difference between adjacent beams of projection is maximal, and at least 80% points of one beam lies in region *O* (i.e. line P_A-P_B is aligned at the boundary of area *O* so that for two adjacent beams of projection one is in area *O* and the neighbouring one is out of area *O*).

7.4. The same steps repeat for line segment P_B-P_C.

7.5. Lines P_A-P_B and P_B-P_C are now aligned to Finder Pattern.

8. Check Timing Pattern

Once Finder Pattern is identified, we must verify if Timing Pattern is present on two opposite sides of the Finder Pattern and the number of rows and columns of the Data Matrix code must be determined.

May be the bounding box does not define exactly borders of the Data Matrix code, so the sequence of black and white modules (the Timing Pattern) must be checked in wider surrounding of the initial bounding box (Fig. 14).

For each shift of the line segment P_C-P_D and P_A-P_D the number of local extremes (extremes that differ more than standard deviation of intensities in gray scale image in Data Matrix region), and the sum of gradients (absolute change of intensity) are computed along the line segment. Position, where sum of gradients is maximal is selected.

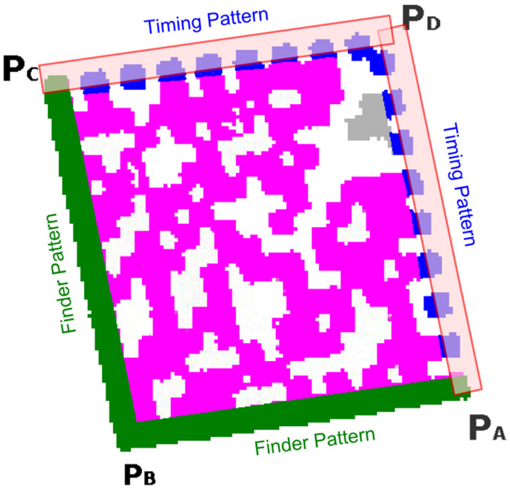


Fig. 14 Areas of Timing Patterns

9. Decode Data Matrix code

Once the number of rows and columns in Data Matrix code is identified, binary matrix can be composed. In this matrix dark modules in image are transformed to 1 values and light modules in image to 0 values. To distinguish dark and light modules in the Data Matrix region in the image, the threshold value T_R is used:

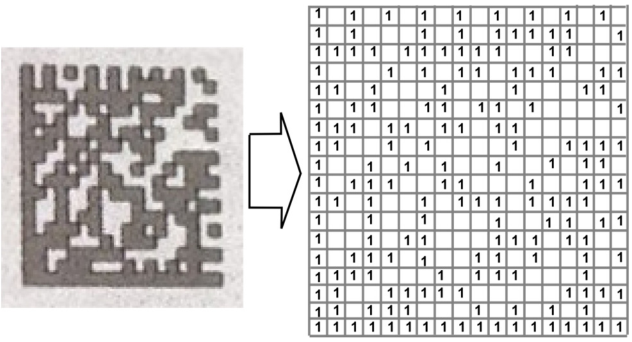
$$T_R = \frac{1}{N}\sum_{i\in region} I_i, \qquad (3)$$

where:

N is the number of all points in the region and *I* is point intensity in gray scale image.

The central point of the module is considered as the decisive point, and the intensity is determined by bilinear interpolation (if the calculated transformed position does not correspond to the integer coordinates in the image).

Open-source library libdmtx [24] is used for the final decoding of the binary matrix, to receive the original text encoded in Data Matrix code (Fig. 15).



1		1		1		1		1		1		1		1		1	
1		1				1		1		1	1	1	1	1			1
1	1	1	1		1	1	1	1	1	1			1	1			
1				1		1		1	1		1	1	1			1	1
1	1		1				1				1				1	1	
1		1	1			1	1		1	1		1					1
1	1	1		1	1		1	1		1	1						
1	1			1		1					1			1	1	1	1
1			1		1		1			1			1		1	1	
1		1	1	1			1	1				1			1	1	1
1	1		1			1		1	1	1		1	1	1	1		
1			1			1				1			1	1		1	1
1			1		1	1				1	1	1		1	1		
1		1	1	1		1			1	1		1			1		1
1	1	1	1				1		1	1	1				1		
1	1			1	1	1	1	1						1	1	1	1
1	1		1	1	1				1		1		1		1		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 15 Transformation of central points of the modules into the binary matrix

RESULTS OF RESEARCH

Presented method has been verified on a two testing sets. Testing set #1 contained together 65 samples of Data Matrix codes, where 21 samples was synthetic Data Matrix codes of various sizes and rotations, 19 samples was Data Matrix codes from Internet images, and 25 samples was Data Matrix codes marked by laser on metal tool. This testing set was designed, to contain high diversity of Data Matrix codes, to verify the robustness of the proposed algorithm. Testing set #2 contained 144 industrial samples taken from a specific manufacturing process.

The results of the proposed method were compared with the results of other available open-source and commercial solutions:

- Google ZXing (open-source) [21],
- Leadtools Data Matrix SDK [22],
- Dynamsoft Barcode Reader SDK [23],
- Libdmtx (open-source) [24],
- DataSymbol Barcode Reader SDK [25],
- Inlite Barcode Reader SDK [26].

In Table 1, 2 and Fig. 16, 17 are compared the results of our proposed algorithm against the results of other algorithms.

Table 1
The number of correctly recognized Data Matrix codes from testing set #1 containing 65 samples

Software	**Synthetic samples a)**	**General samples from the Internet b)**	**Specific industrial samples c)**	**Overall number of recognized samples**
Google ZXing (*open-source*)	1	3	0	4
LEADTOOLS Data Matrix SDK	19	18	6	43
Dynamsoft Barcode Reader SDK	21	18	12	51
libdmtx (*open-source*)	15	18	19	52
DataSymbol Barcode Reader SDK	21	16	19	56
Inlite Barcode Reader SDK	21	18	18	57
Our proposed algorithm	**21**	**19**	**24**	**64**

Table 2
The number of correctly recognized Data Matrix codes from testing set #2 containing 180 samples

Software	**Number of recognized samples**
Dynamsoft Barcode Reader SDK	69
LEADTOOLS Data Matrix SDK	77
DataSymbol Barcode Reader SDK	114
Inlite Barcode Reader SDK	139
libdmtx (*open-source*)	144
Our proposed algorithm	**172**

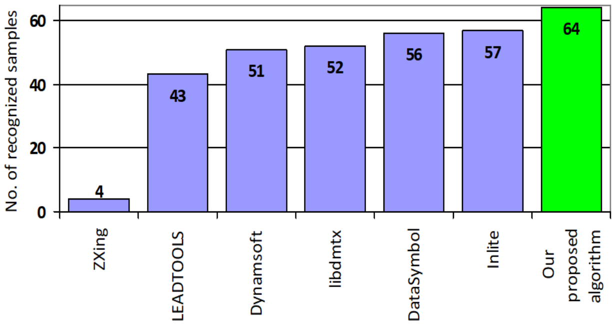


Fig. 16 Comparison of the results achieved by our algorithm with the results of other solutions on testing set #1

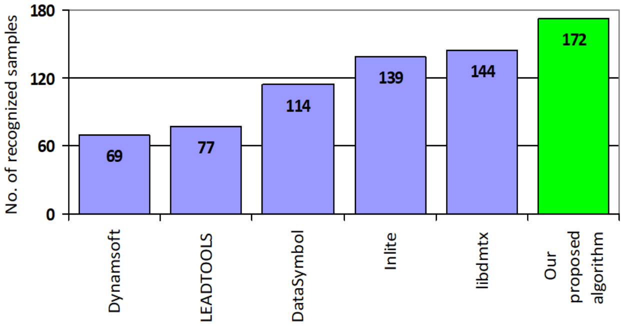


Fig. 17 Comparison of the results achieved by our algorithm with the results of other solutions on testing set #2

As can be seen from Tab. 1, 2 and Fig. 16, 17, our algorithm achieved best results compared to the other solutions. Recognition rate of our algorithm was up to 98% of the Data Matrix codes (64 out of 65) on testing set #1 and up to 95% (172 out of 180) on testing set #2.

DISCUSSION

For successful decoding of 2D codes (especially of smaller dimensions) their exact localization (determination of position, dimensions and orientation in the image) is essential. Sometimes, the difference of one point can make the 2D code impossible to decode (i.e., to read what it contains) even though the 2D code has been localized. That is why robust detection algorithm must deal with precise localization of Data Matrix codes in image.

Experiments have shown that an improvement in recognition can be further achieved by repeating the algorithm on a progressively resized image to 75% and 50% of its original size. In the resized image are suppressed discontinuities caused by adaptive thresholding that cause incomplete identification of the Data Matrix code area.

As can be seen from Fig. 16, our solution has outperformed competitive solutions. We tested competitive software using default settings, mostly by using an online recognizing service that does not allow us to set or tune recognition parameters. However, most competing decoders have closed sources and therefore we cannot compare their detection algorithms in detail or set-up their parameters.

We assume that our proposed algorithm achieved better results mainly because of the use of a modified local thresholding technique and accurate Finder Pattern border determination, by maximizing neighbouring projection difference. In our prior papers we dealt also with several other methods for locating Data Matrix codes in images [11, 12, 13].

In the works of other authors, there are used techniques based on edge detection, on connection of the edge points with similar gradient angle and on the searching of the perpendicular regions [2, 8, 9], Radon transform [5] or Hough transform [10]. QR code recognition is also a point of interest to other authors [7, 15, 16].

CONCLUSION

Nowadays, a new strategy (Smart Specialisation) is emerging, which connects various branches of the industry and uses innovative technological solutions [1]. One of the problems are to be solved is identification of individual items in business processes. With automated identification enabled by Data Matrix codes, an enterprise can take advantage of new technologies that help to streamline business processes, reduce costs, and improve service levels to gain competitive advantage.

As has been stated above, Data Matrix codes can be used to label logistic units, parts, warehousing positions, but also to navigate automated robots in production engineering [3, 6, 17, 20].

In assessing their benefits, their low cost, accuracy, speed, reliability, flexibility and efficiency, as well as the ability to record large amounts of data on a small area, they appear to be ideal means for industrial applications.

In this paper we present our own effective method of Data Matrix code recognition. Our goal was to design a real-time Data Matrix code recognition method that does not require extensive pre-processing, allows automatic processing and evaluation without operator intervention, is invariant to position of the Data Matrix code in the image (shift, rotate, scale), partially compensates for code distortions (light reflection, incomplete code) and poor image quality.

We designed and tested a method for locating and decoding Data Matrix codes in images using Finder Pattern and Timing Pattern, a modified local thresholding technique, joining points to contiguous areas, outer region boundaries. The proposed method was verified on a test set from a real environment and was compared with competing solutions. The experimental results presented above show that our method achieved better results than commercial solutions.

REFERENCES

[1] W. Biały. “The possibilities of production engineering in the development of smart specialisations illustrated with the example of the Silesian Voivodeship”. *Management Systems in Production Engineering.* vol. 4 (24), pp. 252-258, 2016.

[2] J.B. Burns, A.R. Hanson and E.M. Riseman. “Extracting straight lines”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 4, pp. 425-455, 1986.

[3] P. Božek. “Automated detection type body and shape deformation for robotic welding line”. *Advances in Intelligent Systems and Computing*, vol. 240, pp. 229-240, 2014.

[4] D. Bradley, G. Roth. “Adaptive thresholding using the integral image”. *Journal of Graphics Tools*, 12(2), pp. 13-21, 2007.

[5] H. Donghong, T. Hui and C. Xinmeng. “Radon transformation applied in two dimensional barcode image recognition”. *Journal of Wuhan University*, vol. 5, pp. 584-588, 2005.

[6] P. Frankovský, D. Hroncová, I. Delyová, P. Hudák. „Inverse and forward dynamic analysis of two link manipulator“. *Procedia Engineering*, vol. 48, pp. 158-163, 2012.

[7] P. Gaur, S. Tiwari. “Recognition of 2D Barcode Images Using Edge Detection and Morphological Operation”. *International Journal of Computer Science and Mobile Computing*, vol. 3 (4), pp. 1277-1282, 2014.

[8] G.R. Gioi, J. Jakubowicz, J.M. Morel and G. Randall. “LSD: a fast line segment detector with a false detection control”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722-732, 2010.

[9] Q. Huang, W-S. Chen, X-Y. Huang, Y-Y Zhu. (2012, Jan). “Data Matrix Code Location Based on Finder Pattern Detection and Bar Code Border Fitting”. *Mathematical Problems in Engineering*. [On-line]. Available: www.hindawi.com/journals/mpe/2012/515296 [Jun. 27, 2018]

[10] Z. Chenguang, Y. Na and H. Rukun. “Study of two dimensional barcode identification technology based on HOUGH transform”. *Journal of Changchun Normal University*, vol. 4, pp. 94-98, 2007.

[11] L. Karrach, E. Pivarčiová. “The analyse of the various methods for location of Data Matrix codes in images”, in *Elektro* 2018.

[12] L. Karrach, E. Pivarčiová, Y.R. Nikitin. “Comparing the impact of different cameras and image resolution to recognize the data matrix codes”. *Journal of electrical engineering*, No. 4, pp. 286-292, 2018.

[13] L. Karrach, E. Pivarčiová. “Options to use data matrix codes in production engineering”. *Management Systems in Production Engineering*, vol. 26 (4), pp. 231-236, 2018.

[14] P. Koleda, M. Hrčková. “Global and Local Thresholding Techniques for Sawdust Analysis”. *Acta Facultatis Technicae*, vol. 1, pp. 33-42, 2018.

[15] S. Li, J. Shang, Z. Duan, J. Huang. “Fast detection method of quick response code based on run-length coding”. *IET Image Processing*, vol. 12 (4), pp. 546-551, 2018.

[16] J-A. Lin, CH-S. Fuh. “2D Barcode Image Decoding”. *Mathematical Problems in Engineering*, 2013.

[17] M. More, O. Liska, J. Kovac. “Experimental verification of force feedback for rehabilitation robot”. *International journal of engineering research in Africa*, vol. 18, pp. 123-129, 2015.

[18] A. Rosenfeld, J. Pfaltz. “Sequential Operations in Digital Picture Processing” *Journal of the ACM*, vol. 13 (4), pp. 471-494, 1966.

[19] G.T. Schmidt. “SCHMIDT at IMTS: Welcome to the Laser Show.” Internet: https://www.gtschmidt.com/news/schmidt-imts-welcome-laser-show, Aug. 17, 2016 [Jan. 25, 2020].

[20] Y. Turygin, P. Božek, Y. Nikitin, E. Sosnovich, A. Abramov. “Enhancing the reliability of mobile robots control process via reverse validation”. *International Journal of Advanced Robotic Systems*, Vol. 13(6), pp. 1-8, 2016.

[21] Google. „ZXing ("Zebra Crossing") barcode scanning library for Java, Android”, Internet: github.com/zxing, [Apr. 12, 2018].

[22] Leadtools. „Data Matrix SDK”, Internet: www.leadtools.com/sdk/barcode/2d-datamatrix, [Apr. 12, 2018].

[23] Dynamsoft. „Barcode Reader SDK”, Internet: www.dynamsoft.com/Products/Dynamic-Barcode-Reader.aspx, [Apr. 12, 2018].

[24] M. Laughton. „Open source software for reading and writing Data Matrix barcodes“, 2011, Internet: github.com/dmtx, [Apr. 12, 2018].

[25] DataSymbol. „DataSymbol Barcode Reader SDK”, Internet: www.datasymbol.com/barcode-reader-sdk/barcode-reader-sdk-for-windows/online-barcode-decoder.html, [Jan. 17, 2020].

[26] Inlite Research Inc. „Barcode Reader SDK”, Internet: online-barcode-reader.inliteresearch.com, [Apr. 12, 2018].

Ladislav Karrach
Technical University in Zvolen
Faculty of Technology
Department of Manufacturing
and Automation Technology
Študentská 26, 960 53 Zvolen, Slovak Republic
e-mail: karrach@zoznam.sk

Elena Pivarčiová*
Technical University in Zvolen
Faculty of Technology
Department of Manufacturing
and Automation Technology
Študentská 26, 960 53 Zvolen, Slovak Republic
Tel: +421 903 140 405
e-mail: pivarciova@tuzvo.sk
*corresponding author