International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:6, No:4, 2012

Comparison of Hough Transform and Mean Shift Algorithm for Estimation of the Orientation Angle of Industrial Data Matrix Codes

Ion-Cosmin Dita, Vasile Gui, Franz Quint, and Marius Otesteanu

Abstract—In automatic manufacturing and assembling of mechanical, electrical and electronic parts one needs to reliably identify the position of components and to extract the information of these components. Data Matrix Codes (DMC) are established by these days in many areas of industrial manufacturing thanks to their concentration of information on small spaces. In today's usually order-related industry, where increased tracing requirements prevail, they offer further advantages over other identification systems. This underlines in an impressive way the necessity of a robust code reading system for detecting DMC on the components in factories. This paper compares two methods for estimating the angle of orientation of Data Matrix Codes: one method based on the Hough Transform and the other based on the Mean Shift Algorithm. We concentrate on Data Matrix Codes in industrial environment, punched, milled, lasered or etched on different materials in arbitrary orientation.

Keywords—Industrial Data Matrix Code; Hough Transform; Mean Shift

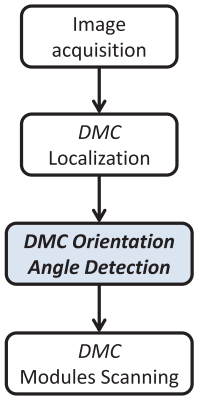


Fig. 1. Identification process of Data Matrix Code

I. Introduction

THE industrial Data Matrix Code (DMC) is a two-dimensional matrix bar-code consisting of dots (modules) arranged in a square. The information to be encoded can be text or raw data [1]. Error correction codes are added to increase the robustness of the code: even if they are partially damaged, they can still be read. As more data is encoded in the symbol, the number of modules (rows and columns) increases from 8×8 to 144×144.

For industrial purposes, Data Matrix Codes are marked directly on the parts by different techniques like milling, punching, lasering or etching [1].

Fig. 1 gives an overview of the identification process of Data Matrix Code, focusing on the image processing steps to find the angle of orientation of the DMC. After image acquisition, the process starts with the localization of the Data Matrix Code. As result, a region of interest (ROI) is obtained, in which the DMC is located. In order to reliably detect the angle of orientation, a set of modules of the DMC having good quality is selected. Using these candidate modules, the angle of orientation of the DMC is estimated. Therefore we provide two methods: one based on the Hough Transform, the other based on the Mean Shift Algorithm. The purpose of this paper is to compare the two methods and to point out their strengths and weaknesses. Finally knowing the location and the orientation of the DMC, the process continues by scanning the modules and reading the code.

I. C. Dita, V. Gui and M. Otesteanu - Faculty of Electronics and Telecommunications, Politehnica University of Timisoara, Romania, Email: cosmin.dita@etc.upt.ro, vasile.gui@etc.upt.ro, marius.otesteanu@etc.upt.ro
F. Quint - Faculty of Electrical Engineering and Information Technology, University of Applied Sciences Karlsruhe, Germany, Email: franz.quint@hs-karlsruhe.de

II. DMC Localization

This block is used to determine a Region of Interest *(ROI)* where the Data Matrix Code lies. To this end, the real world size of the DMC and camera's focal length are considered to be approximately known. The image is thresholded using an adaptive threshold level [2] [3].

To reliable detect the Data Matrix Code it is convenient to have a solid shape rather than separated modules. In this work, the morphological close operation [4] is used to this end. The first stage of the close operation dilates the foreground in order to fill the empty spaces between modules, whereas the second stage, erosion, restores the original size of the DMC area. The dilation operation expands the objects. By choosing an appropriate structural element, after dilation all modules will be connected, building a square.

Using the erode operation, the objects are resized to the initial dimension. The effect generated by the eroding operation is to thin the objects. The two operations performed in sequel are called Image Closing.

For each object in the image, by taking theirs maximum and the minimum coordinates define the four enclosing rectangle. Using the position vectors of these points (Fig. 2), the orientation angle of the Data Matrix Code is given by:

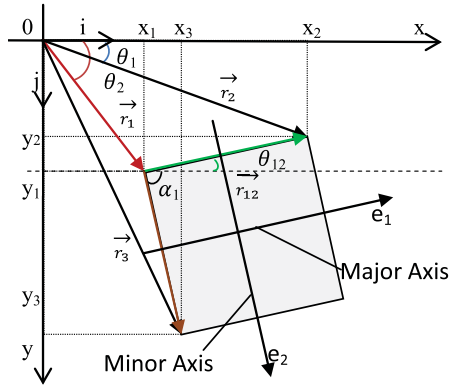


Fig. 2. Corner angles, Major and Minor axis calculation

$$\cos\alpha_1 = \frac{\overrightarrow{r_1}\cdot\overrightarrow{r_2}}{|\overrightarrow{r_1}|\cdot|\overrightarrow{r_2}|}. \quad (1)$$

The angle α and the major axis of each object are calculated using the geometrical moments as given in eq. (2, 3).

$$\mu_{pq} = \sum\sum(x-x_c)^p\cdot(y-y_c)^q\cdot f(x,y), \quad (2)$$

$$\alpha = \frac{1}{2}\cdot\arctan\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}, \Rightarrow y = (x-x_c)\tan\alpha + y_c. \quad (3)$$

The angle corners α_i together with other features, like the area of the solid square, are used to locate the region of interes which contains a valid DMC.

By using the vectors r_1 and r_2 one can calculate the orientation angle of the DMC. However, due to the image closing operation, this angle is not obtained with the required accuracy. Even small errors in the angle can mislead the modules scanning process from one row into the adjacent one.

III. DMC Angle Estimation Based on Hough Transform Algorithm

A. Hough Transform

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing [5]. The purpose of this technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.

The simplest case of Hough transform is the linear transform for detecting straight lines. In the image space, the straight line can be described as $y = mx + b$ and can be graphically plotted for each pair of image points (x, y). In the Hough transform, a main idea is to consider the characteristics of the straight line not as image points (x_1, y_1), (x_2, y_2), but instead, in terms of its parameters, the slope parameter m and the intercept parameter b. Based on that fact, the straight line $y = mx + b$ can be represented as a point (b, m) in the parameter space. However, one faces the problem that vertical lines give rise to unbounded values of the parameter m. For computational reasons, it is therefore better to use a different pair of parameters, denoted r and θ, for the lines in the Hough transform.

The parameter r represents the distance between the line and the origin, i.e. the length of a vector from the origin perpendicular to the line, while θ is its angle. Using this parametrization, the equation of the line can be written as: [5]

$$y = -\frac{cos\theta}{sin\theta}\cdot x + \frac{r}{sin\theta}, \quad (4)$$

which can be rearranged to $r = x\cdot cos\theta + y\cdot sin\theta$.

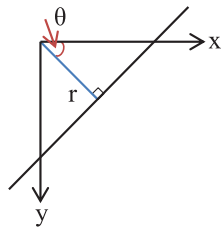


Fig. 3. r and θ line parametrization

The Hough transform algorithm uses an array, called accumulator, to detect the existence of a line $r = xcos(\theta) + ysin(\theta)$. The dimension of the accumulator is equal to the number of unknown parameters of the Hough transform problem. The linear Hough transform problem has two unknown parameters: r and θ. The two dimensions of the accumulator array would correspond to quantized values for r and θ. For each pixel and its neighborhood, the Hough transform algorithm determines if there is enough evidence of an edge at that pixel. If so, it will calculate the parameters of that line, and then look for the accumulator's bin that the parameters fall into, and increase the value of that bin. By finding the bins with the highest values, typically by looking for local maxima in the accumulator space, the most likely lines can be extracted, and their (approximate) geometric definitions read off.

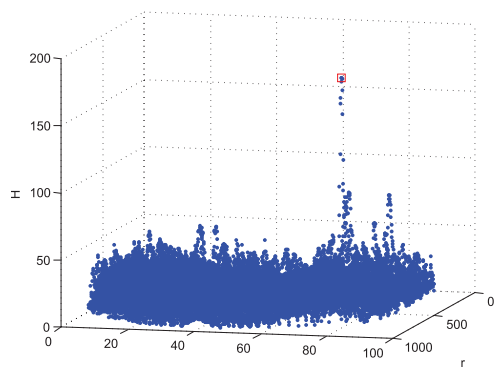


Fig. 4. Hough transform accumulator

The simplest way of finding these peaks is by applying some form of threshold, but different techniques may yield better results in different circumstances - determining which lines are found as well as how many. Since the lines returned do not contain any length information, it is necessary to find which parts of the image match up with which lines.

B. Hough Transform implementation for DMC angle estimation

Let's consider the Data Matrix Code binary image from Fig. 5.

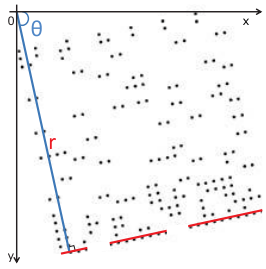


Fig. 5. The orientation angle

The Hough Transform steps for Data Matrix Code angle computation are:

- For each data point, a number of lines going through it at different angles are generated. A plot of such lines is illustrated in 5.
- The length, r and angle, θ, of each dashed line is measured as is shown in Fig. 4.
- This is repeated for each data point.
- In the accumulator matrix of the Hough-Transform (Fig. 6), by searching the maxima the lines are detected. From the same figure we can associate the angle and the distance corresponding to the maxima of the Hough accumulator. The maximum found in Fig. Fig. 6 is marked by a white square.

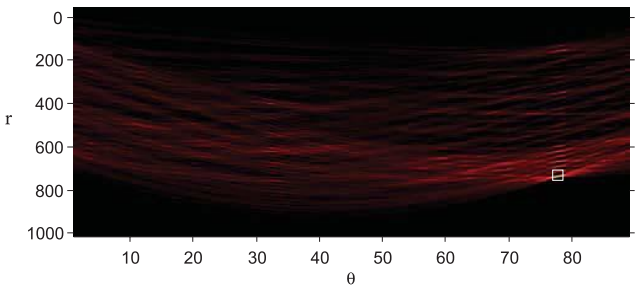


Fig. 6. The Hough Transform of the Data Matrix Code image

IV. DMC ANGLE ESTIMATION BASED ON MEAN SHIFT ALGORITHM

A. Mean Shift

Mean shift is a non-parametric feature-space analysis technique, a so-called mode seeking algorithm. It locates the maxima of a density function given discrete data sampled from that function [6]. By iteratively moving an analysis window starting from an initial estimate x, it detects the modes (maxima) of the density.

Let a kernel function $K(x_i - x)$ be given. This function determines the weight of nearby points for re-estimation of the mean. Typically one uses the Gaussian kernel on the distance to the current estimate, $K(x_i - x) = e^{c||x_i - x||^2}$. The weighted mean of the density in the window determined by K is:

$$y = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}, \tag{5}$$

where $N(x)$ is the neighborhood of x, a set of points for which $K(x) \neq 0$. The mean-shift algorithm now sets $x \leftarrow m(x)$, and repeats the estimation until $m(x)$ converges, Fig. 7 [7].

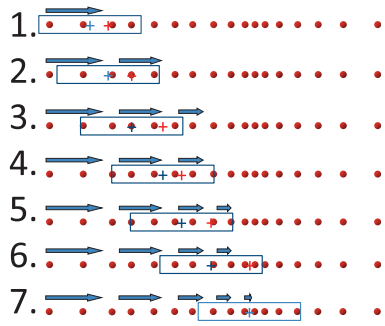


Fig. 7. Mean Shift Algorithm

If dense regions (or clusters) are present in the feature space, then they correspond to the mode (or local maxima) of the probability density function. For each data point, Mean shift associates it with the nearby peak of the data-set's probability density function. For each data point, Mean shift defines a window around it and computes the mean of the data point. Then it shifts the center of the window to the mean and repeats the algorithm till it converges. After each iteration, one can consider that the window shifts to a more denser region of the data-set.

B. Mean Shift implementation for DMC angle estimation

Let's consider the labeled image of a Data Matrix Code and all the coordinates of the objects from the image. One starts from the first module to the last one from the image and calculates the angle of the straight line created by the modules, using the equation:

$$\alpha_{ij} = \arctan \frac{x_j - x_i}{y_j - y_i}, \tag{6}$$

where (x_i, y_i) şi (x_j, y_j) are the coordinates of modules i and j.

The calculated angels are stored in a connected list in increasing order.

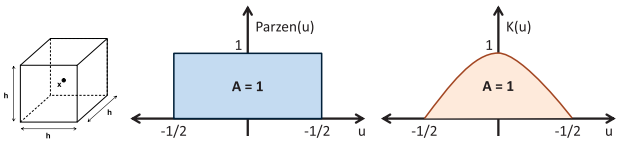
The analysis window's size should not be bigger than half of the angles. In the example from Fig. 9(b) the size is chosen as half of the space taken by the angles' value.

The general expression for the non-parametric density estimation is:

$$P(x) \cong \frac{k}{N \cdot V}, \tag{7}$$

where V is the volume of the x position, N is the total number of the elements, and k is the number of elements from inside V [8].

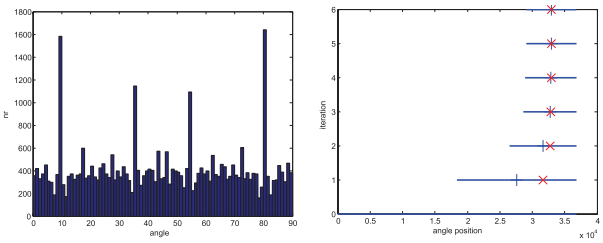
If we assume that the region surrounding the examples k is a Hypercube with sides of length h, centered in an estimated point x, its volume is given by $V = h^D$, where D is the number of dimensions, Fig.8.



(a) V_D-Space (b) Analysis window (c) Kernel function

Fig. 8. Analysis window

In our case the dimension $D = 1$, this means that V is the length of the segment, and k is the number of the values contained in the segment [9] [10].



(a) General angles histogram (b) Mean Shift iterations

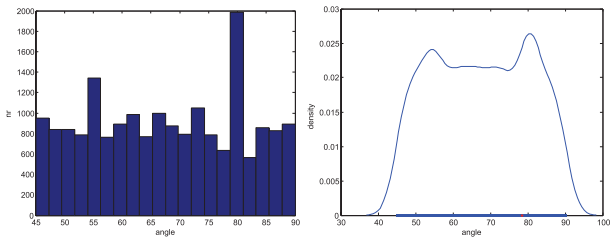
Fig. 9. Mean Shift Algorithm for angle computation

Next, based on Fig. 9(b), the Mean Shift steps are described:

- In the first step, the analysis segment is centered in the center of the data base coverage. The length of the segment is equal with half of the coverage. Using the equation (7), the density center of the points from the segment is computed. The center of the segment is moved in the position calculated by the equation, Fig. 9 (b),
- In the second step, the segment would exceed the coverage of the angels. Because of that, its dimension is recalculated using the new values form the segment and its real length.
- The procedure is repeated until it converges to mean, $3^{th} - 6^{th}$ iterations.

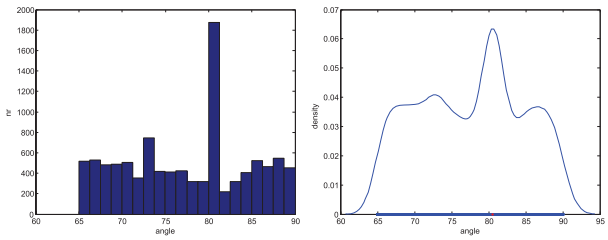
In Fig. 10 - 13 the first five iterations are shown, the 6th iteration being the same with the 5th. On these two last iterations the algorithm converges and stops.

We can notice that the peak position of the function densities stabilizes after each iteration. The Mean Shift does not require a pre-parametrization process, the algorithm is adapting according to the values of the area subject to estimation.



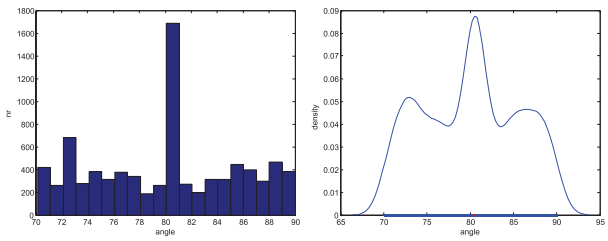
(a) Angles histogram (b) The weighted average

Fig. 10. Mean Shift - 1th iteration



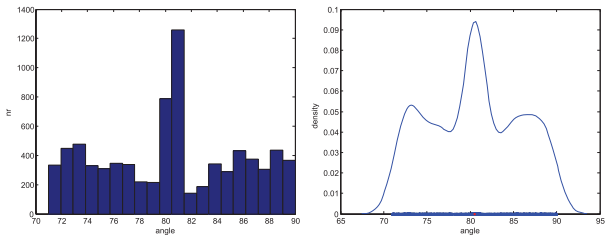
(a) Angles histogram (b) The weighted average

Fig. 11. Mean Shift - 2th iteration



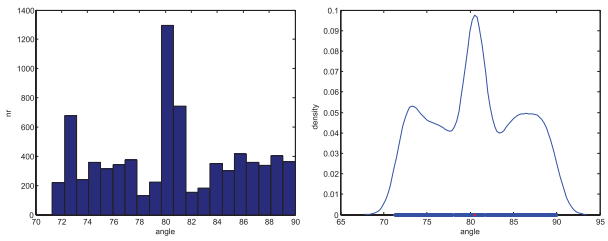
(a) Angles histogram (b) The weighted average

Fig. 12. Mean Shift - 3th iteration



(a) Angles histogram (b) The weighted average

Fig. 13. Mean Shift - 4th iteration



(a) Angles histogram (b) The weighted average

Fig. 14. Mean Shift - 5th iteration

V. Comparison of Hough Transform and Mean Shift Algorithm for DMC angle estimation

This chapter compares two methods for estimating the angle of orientation of the Industrial Data Matrix Codes. The first method is based on Hough Transform and the second is based on the Mean Shift algorithm [11] [7]. The purpose of this chapter is to highlight strengths and weaknesses of the two methods presented in Chapters III, IV.

A. Experiment

In order to compare the two methods one should analyze more DMC patterns at different angles to have a satisfying statistic. This would be more difficult based on real objects marked with Data Matrix Code. In this case the human error can occur. Therefore, we develop a test method based on synthetic - generated DMC patterns.

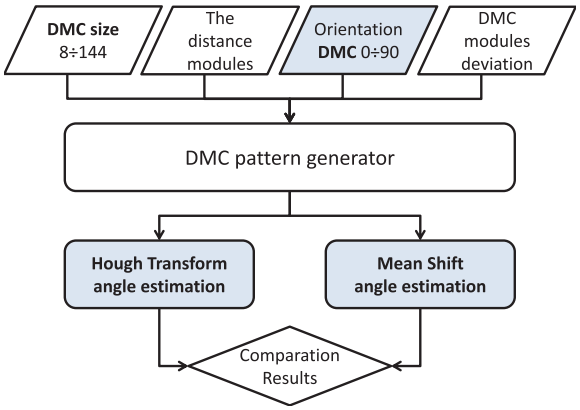


Fig. 15. The description of the experiment

Knowing the structure of a real DMC pattern, its characteristics like: the size of the pattern, the distance between the modules and its angle one can generate a synthetic image of the pattern (Fig. 15). If this process is automated by generating successive patterns of several different angles, then we can simultaneously test the two methods and achieve a statistics of their results. When an image is acquired, errors can occur. This error may be caused by the optical system of the camera, by its analog-digital converter, or by external factors: light, noise, camera lenses, etc. The image generated by the generator of the DMC pattern proposed in this chapter, would be a perfect image without the intervention of the above enumerated factors. Therefore, as for the the synthetic - generated DMC pattern to be similar with the real pattern acquired with a video camera, a position error is introduced, "σ". This error, "σ", moves the centers of the DMC modules, randomly between two limits imposed manually.

If we consider two neighboring modules "1" and "2", located at a distance "d" between them (Fig. 16), the maximum error E_{max} should not be greater than half of the distance between the modules. In one dimension, we can associate this error with Gaussian normal distribution, whose density of probability function is written (eq. 8) [12]:

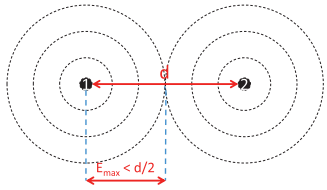


Fig. 16. The maxim deviation of a DMC module

$$f(X_m) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(X_m-\overline{X})^2}{2\sigma^2}} \tag{8}$$

for: $-\infty < X_m - \overline{X} < \infty$, where : $\overline{X}$ is the average of the input and σ - the square average deviation.

In Fig. 17 the shape of the probability function in two dimensions it is presented, It can express the distribution of the error in the image space.

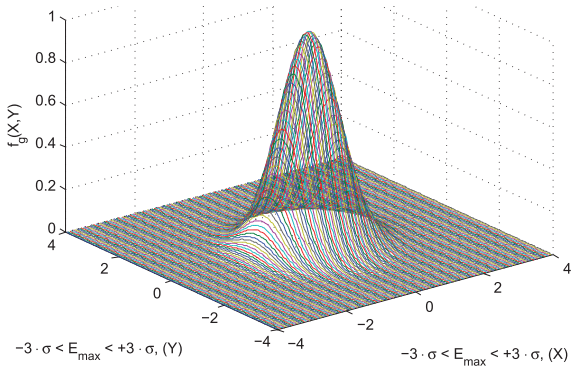


Fig. 17. Gaussian distribution curve of probability density of the error in image space

Knowing all these, one can generate the Data Matrix pattern, creating a network of points using the relationship :
for the rows:

$$\begin{cases} x_1 = (x_0 + d \cdot \cos\alpha) + E_\sigma \\ y_1 = (y_0 + d \cdot \sin\alpha) + E_\sigma \end{cases}, \tag{9}$$

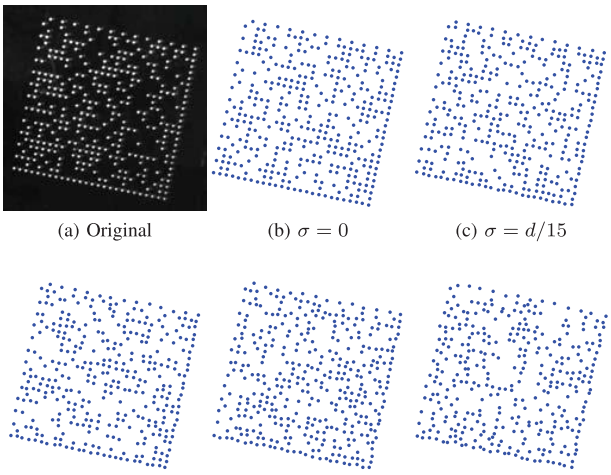
for the columns:

$$\begin{cases} x_2 = (x_0 - d \cdot \sin\alpha) + E_\sigma \\ y_2 = (y_0 + d \cdot \cos\alpha) + E_\sigma \end{cases}, \tag{10}$$

where α is the orientation angle of the pattern, d is the distance between the modules and E_σ is the error of the position of the DMC modules, a random value supplied according to the density of probability (eq. 8).

The probability of density parameters are chosen for $\overline{x} = 0$, so that by $(x_0 + d \cdot cos\alpha)$, it can already choose the central position. The distribution has to provide the position from the correct deviation. And because $E_{max} < d/2$ and as in the limits $[\overline{x} - 3\sigma, \overline{x} + 3\sigma]$, are included $99,7\%$ of all values, we choose $3\sigma = E_{max}$, then $E_{max} < d/6$.

Fig. 18 displays the original Data Matrix Code image and the synthetically pattern built on different levels of error. Note that the DMC image is increasingly distorted as the average quadratic deviation is higher.



(a) Original (b) $\sigma = 0$ (c) $\sigma = d/15$

(d) $\sigma = d/12$ (e) $\sigma = d/9$ (f) $\sigma = d/6$

Fig. 18. The DMC pattern synthetically generated, $Size = 26 \times 26$, $d = 11px$, $\alpha = 78^o$

To test the methods, they must be parameterized in equal conditions, at an optimal scale, so the results can be compared to each other. Next, we briefly present the two methods and the factors that directly influence algorithms proposed for the test.

Hough Transform

We know from Chapter III, that the Hough Transform algorithm uses a matrix called accumulator to detect the existence of a line: $r = x \cdot cos(\theta) + y \cdot sin(\theta)$. The matrix size accumulator match the measured values for r and θ, and station columns. For each pixel and its neighborhood, the Hough Transform algorithm determines whether there is sufficient evidence for that pixel [13]. The parameter which depends on the Hough Transformation algorithm is the number columns from the accumulator. The more columns the accumulator has, the precise the accumulator is, but more instable in the same time. However, the fewer columns the accumulator of the transformation has, the stable the algorithm is, but not so precise.

Mean Shift algorithm

From Chapter IV, we know that the Mean Shift algorithm locates the maximum of a density function given by a set of discrete data. Moving iterative and analyzing a Parzen window, one can start from an initial estimated x, and detect the density's peak points [14]. Parameters on which the Main Shift Algorithm depends are the Parzen window (in our case segment) and the number of data falling in this window.

With all this knowledge about the methods that are to be tested, we propose to choose the size of Hough Transformation accumulator. We know that the angles range is $0^o - 90^o$. To have a 1° precision in estimation, we build the accumulator from 90 bins, that means one angle for each bin (90 angles). If we build a higher accumulator, for example: with 900 bins, then the Hough Transform algorithm precision is of $0,1^o$. To have a bigger precision, as for example 9000 bins, the precision is $0,01^o$, that is very good. The DMC generator constructs patterns which their modules have a σ deviation. To find which is the optimal accumulator dimension, we test the follow sizes : $[90, 900, 9000]$.

For the Mean Shift algorithm, we choose the segment "Parzen" half of the data set. If all values from the Parzen segment are chosen by the algorithm, than the accurate is maximal, but on the other hand the processing time is bigger. A higher process speed can be reached only by decreasing the accumulator size. For the experiment, the number of data from the segment is chosen between: 1%, 10% and 100% of the amount of data that fall in the Parzen segment. There are also tested the methods proposed for the four types DMC patterns, according to the modules' deviation $\sigma = [d/15, d/12, d/9, d/6]$. The algorithms are tested in parallel to each situation, there being realized 100 tests. The test results are represented in Tables I - III.

TABLE I
EXPERIMENT 1, PARZEN SEGMENT = 100 VALUES, HOUGH ACCUMULATOR = 90 BINS

Comparison of Hough Transform and Mean Shift algorithm for DMC angle estimation									
Test no.	DMC (E_{max})	Average error $\lvert\alpha_{DMC}-\alpha_{Masurat}\rvert$		E_{min}		E_{max}		Failed estimations	
M-S & Hough		M-S	Hough	M-S	Hough	M-S	Hough	M-S	Hough
1	d/15	0.4644	0.9982	0.0045	0.0071	10.02	15.92	0	1
2	d/12	2.5161	9.0203	0.0116	0.1591	20.94	22.20	0	9
3	d/9	2.6119	9.4872	0.0341	0.5588	22.28	22.32	0	13
4	d/6	3.1246	9.3913	0.0459	0.4114	19.44	22.02	0	18

TABLE II
EXPERIMENT 1, PARZEN SEGMENT = 1000 VALUES, HOUGH ACCUMULATOR = 900 BINS

Comparison of Hough Transform and Mean Shift algorithm for DMC angle estimation									
Test no.	DMC (E_{max})	Average error $\lvert\alpha_{DMC}-\alpha_{Masurat}\rvert$		E_{min}		E_{max}		Failed estimations	
M-S & Hough		M-S	Hough	M-S	Hough	M-S	Hough	M-S	Hough
1	d/15	0.3666	1.1545	0.0044	0.0064	4.750	17.80	0	0
2	d/12	2.7428	6.3865	0.0038	0.1235	21.79	21.30	0	6
3	d/9	2.2995	8.6701	0.0290	0.1268	21.16	22.39	0	13
4	d/6	3.3091	7.9894	0.0818	0.3313	21.72	21.98	0	11

TABLE III
EXPERIMENT 1, PARZEN SEGMENT = 10000 VALUES, HOUGH ACCUMULATOR = 9000 BINS

Comparison of Hough Transform and Mean Shift algorithm for DMC angle estimation									
Test no.	DMC (E_{max})	Average error $\lvert\alpha_{DMC}-\alpha_{Masurat}\rvert$		E_{min}		E_{max}		Failed estimations	
M-S & Hough		M-S	Hough	M-S	Hough	M-S	Hough	M-S	Hough
1	d/15	0.5652	0.8208	0.0056	0.0093	8.488	22.40	0	0
2	d/12	2.7433	6.6697	0.0156	0.1460	21.95	22.19	0	8
3	d/9	2.5994	8.0336	0.0277	0.4943	20.89	22.29	0	11
4	d/6	2.7005	9.5836	0.0013	0.8312	19.45	22.44	0	10

B. Results

In the case of the Hough Transform, the greater the modules' deviation is, regardless of the number samples from the accumulator, the greater the error and the detection angle. In the case of the Mean Shift algorithm, Data Matrix Code modules deviation influences very little the estimated angle. Fig. 19 illustrates the average error for the estimated angles by the tested methods.

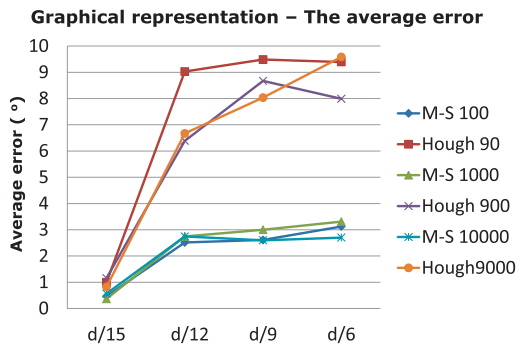


Fig. 19. The average error

VI. Conclusion

The Hough Transform is only efficient if a high number of votes fall in the right bin, so that the bin can be easily detected amid the background noise. This means that the bin must not be too small, or else some votes will fall in the neighboring bins, thus reducing the visibility of the main bin. If we analyze the experiments results, Tables (I - III), can observe that in the case when the accumulator has 90 bins, the algorithm is very stabile (Table I). Following the graphs from the Fig. 19 (up side), we can see that the average error achieve up to $9,39^o$. Also, when the number of parameters is large the average number of votes cast in a single bin is very low, and those bins corresponding to a real figure in the image do not necessarily appear to have a much higher number of votes than their neighbors. In the case when the accumulator size is bigger and the modules deviation is bigger, the algorithm is instable, the average error being $9,58^o$ (Table II). Finally, much of the efficiency of the Hough Transform is dependent on the quality of the input data. Use of the Hough Transform on noisy images is a very delicate matter.

The Mean Shift algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. It requires the bandwidth parameter h to be tuned. The choice of bandwidth is influences convergence rate and the number of clusters. A large h might result in incorrect clustering and might merge distinct clusters. A very small h might result in too many clusters. Mean shift might not work well in higher dimensions. In higher dimensions, the number of local maxima is pretty high and it might converge to a local optima soon, but because in our case the angles set is one dimension the algorithm works with very well results. If we check the Fig. 19 (down side), we can conclude that the average error is independent by the modules deviation or by the bandwidth of Mean Shift Algorithm. The maximal average error is only $3,3^o$, Table I.

Being independent of the choice of some in width, in our case the Mean shift algorithm provided us better results in the estimate of the angle of orientation of the Industrial Data Matrix Code.

Acknowledgment

This work was partially supported by the strategic grant POSDRU 6/1.5/S/13, (2008) of the Ministry of Labour, Family and Social Protection, Romania, co-financed by the European Social Fund - Investing in People.

References

[1] INTERNATIONAL STANDARD, "Information technology — International symbology specification — Data matrix."

[2] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man and Cybernetics, DOI - 10.1109/TSMC.1979.4310076 (Systems, Man and Cybernetics, IEEE Transactions on)*, vol. 9, no. 1, pp. 62–66, 1979.

[3] Ye Zhang, Hongsong Qu, and Yanjie Wang, "Adaptive Image Segmentation Based on Fast Thresholding and Image Merging," *Artificial Reality and Telexistence–Workshops, 2006. ICAT '06. 16th International Conference on*, pp. 308–311, 2006.

[4] A. Bovik, Ed., *The Essential Guide to Image Processing*. Elsevier, 2009.

[5] R. O. Duda and P. E. Hart, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Comm. ACM, January, 1972.

[6] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis: Pattern Analysis and Machine Intelligence, IEEE Transactions on," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[7] Yizong Cheng, "Mean shift, mode seeking, and clustering: Pattern Analysis and Machine Intelligence, IEEE Transactions on," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 8, pp. 790–799, 1995.

[8] Rob L. Hyndman, Xibin Zhang, and Maxwell L. King, "Bandwidth Selection for Multivariate Kernel Density Estimation Using MCMC," 2004.

[9] V. C. Raykar and D. R., "Fast optimal bandwidth selection for kernel density estimation: Proceedings of the sixth SIAM International Conference on Data Mining," *Proceedings of the sixth SIAM International Conference on Data Mining*, pp. 524–528, 2006.

[10] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking: Pattern Analysis and Machine Intelligence, IEEE Transactions on," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[11] Ezio Malis and Eric Marchand, "Experiments with robust estimation techniques in real-time robot vision: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on DOI - 10.1109/IROS.2006.282572," *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 223–228, 2006.

[12] H. Guo, "A Simple Algorithm for Fitting a Gaussian Function [DSP Tips and Tricks]: Signal Processing Magazine, IEEE," *Signal Processing Magazine, IEEE*, vol. 28, no. 5, pp. 134–137, 2011.

[13] J. Princen, H. K. Yuen, J. Illingworth, and J. Kittler, "A comparison of Hough transform methods: Image Processing and its Applications, 1989., Third International Conference on: Image Processing and its Applications, 1989., Third International Conference on DOI -," *Image Processing and its Applications, 1989., Third International Conference on*, pp. 73–77, 1989.

[14] Yefeng Zheng, B. Georgescu, Haibin Ling, S. K. Zhou, M. Scheuering, and D. Comaniciu, "Constrained marginal space learning for efficient 3D anatomical structure detection in medical images: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on DOI - 10.1109/CVPR.2009.5206807," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 194–201, 2009.