

# Comprehensive Performance Evaluation of Various Feature Extraction Methods for OCR Purposes

Dawid Sas<sup>1(✉)</sup> and Khalid Saeed<sup>2</sup>

<sup>1</sup> Faculty of Physics and Applied Computer Science,  
University of Science and Technology, Krakow, Poland  
dawsas@wp.pl

<sup>2</sup> Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland  
k.saeed@pb.edu.pl

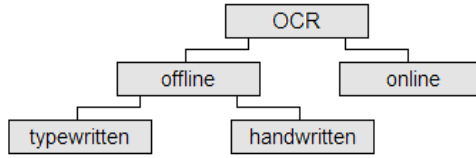
**Abstract.** Optical Character Recognition (OCR) is a very extensive branch of pattern recognition. The existence of super effective software designed for omnifont text recognition, capable of handling multiple languages, creates an impression that all problems in this field have already been solved. Indeed, focus of research in the OCR domain has constantly been shifting from offline, type-written, Latin character recognition towards Asiatic alphabets, handwritten scripts and online process. Still, however, it is difficult to come across an elaboration which would not only cover the topic of numerous feature extraction methods for printed, Latin derived, isolated characters conceptually, but which would also attempt to implement, compare and optimize them in an experimental way. This paper aims at closing this gap by thoroughly examining the performance of several statistical methods with respect to their recognition rate and time efficiency.

**Keywords:** OCR · Feature extraction · Shape descriptors

## 1 Introduction

A simple taxonomy of OCR systems can be presented schematically as in Fig. 1. Online OCR concentrates on recognition of handwriting in real time. It employs digital devices like electronic pads and pens, which allow to acquire data and extract information not only from sheer shape, but also from dynamics of writing. The offline counterpart processes only static contents – the form of glyphs. It can be further divided based upon nature of text it operates on. Heavy variations of style and possible character overlapping account for the main problems related to handwritten OCR, especially while handling cursive scripts, like Arabic alphabet. OCR designed for machine text does not have to handle these issues as the characters are usually well separated and have uniform or at least highly predictable shape. Handwritten subsystems, both offline and online, are often referred to as Intelligent Character Recognition (ICR).

Offline OCR processing consists of the following stages. In the beginning, image is acquired by scanning or taking a photo of a document. Then it undergoes preprocessing. The term encompasses a set of techniques aiming to improve the quality of



**Fig. 1.** Taxonomy of OCR systems

the scan. The next stage is binarization to convert the grayscale image to the form that features only two intensity levels. Binarization is followed by segmentation of lines, words and finally single glyphs or glyph fragments. As characters have been isolated, they are subject to feature extraction which means encoding a shape in a sort of numerical representation. Based upon the features, each glyph is subsequently classified, or labelled, as a member of one of predefined classes. At the final stage of OCR, the whole text is examined word by word against lexical and syntactical compliance with the given language rules.

### 1.1 Approaches to Feature Extraction

There are two main approaches to the feature extraction process: statistical and structural. Statistical methods transform a shape into a strictly ordered set of specified length, the so called *feature vector*, which represents a point in multidimensional *feature space*. They cannot work without a training set, serving as a database of mappings between cases and corresponding classes. Based upon the set, a decision is made into which class a new, hitherto unknown, case should be incorporated. On the other hand, structural approach is directed towards decomposing shapes into simpler pieces and establishing relationships between them. Detailed description of structural techniques can be found in [1].

### 1.2 Criteria of Statistical Coding Techniques

The attributes of a well-performing statistical feature extraction technique include capability of grouping all instances of the same class into tight clusters in feature space in order to aid classification. In [2] this is formulated as “*minimizing the within class pattern variability while enhancing the between class pattern variability*”. Also, high robustness to noise and distortions as well as invariance to geometric transformations are demanded. The code format should be compact which means shape information to be carried by as few features as possible. Dimensionality reduction is essential as to guarantee that the computational expense of classification process does not exceed acceptable values. The same remark, regarding execution speed, applies as well to the extraction process algorithm.

### 1.3 Motivation

One of the most extensive surveys concerning statistical feature extraction methods for OCR purposes was covered in [3]. The authors put considerable efforts to set together multiple techniques, considering perspectives of their application to different

forms of glyphs. They also discussed the aspects of feature invariance and reconstructibility of the shape from a descriptor. The researchers, however, did not show any comparison of the described methods in an experimental way. The present paper is prepared with a view to complement the aforementioned survey with relevant tests. Numerous description techniques were also listed out by [4] (Arabic handwriting) and [5] (Devanagari script).

The paper is organized as follows. The review of shape description techniques is conducted within Section 2. Section 3 explains the scope of the research. Section 4 gives the results of the tests. Finally, the work is summarized in Section 5.

## 2 Feature Extraction Techniques

Within the family of statistical descriptors one can distinguish classes utilizing concepts like [6]:

- pixel distribution (i.e. zoning, crossings, projection methods),
- moment invariants (i.e. central moments, Hu moments),
- series expansion (i.e. Zernike moments),
- unitary transforms (i.e. Fourier transform, Hadamard transform, cosine transform),
- contour description (i.e. Freeman chaincodes, polyline approximation, elliptic Fourier descriptors).

There are other approaches to the concept of feature extraction techniques like the method based on Toeplitz matrix minimal eigenvalues for script feature extracting and description [7, 8] or soft computing approaches. In this paper, however, the authors have limited their research to the most relevant methods and algorithms.

### 2.1 Zoning

Glyph bounding box is divided into rectangular areas. A parameter is computed inside each rectangle and treated as a single feature (Fig. 2a). The authors of [9] partition an image of size 60×90 into 54 10×10 squares. Pixels along each of the 19 diagonals of a zone are summed up and the amounts are eventually averaged. Further, average values of the zones stacked horizontally and vertically contribute to extra 15 features. In this paper the image is zoned as suggested in [9] and pixel density in each region, row and column serves as a quantity.

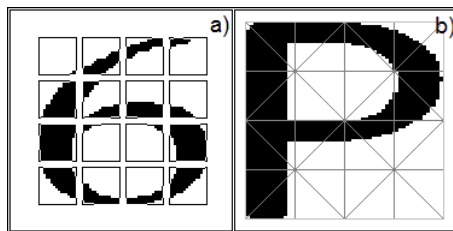


Fig. 2. a) Zonal division, b) Grid used in crossings technique

### 2.2 Crossings

A custom grid of lines is superimposed on the image and the spots where lines meet glyph pixels are used to form features. In the authors' implementation images are scaled to  $(4n + 3) \times (4n + 3)$  squares ( $n \in \mathbb{N}$ ) and quartered so that one-pixel-wide gap is left between the pieces. Four lines are stretched through each quarter: one horizontal, one vertical and two oblique, each going through the center. Additional four sections run from the middle of the image, orthogonally towards its edges (Fig. 2b). Along each line pixels are registered and their positions are averaged. The obtained pairs of numbers are dependant on each other and hence only one value is selected towards the feature vector. Thus, vector dimensionality is the same as the number of lines – 20. To prevent the algorithm from getting stuck due to undefined situations, several emergency scenarios must be taken into account.

### 2.3 Projection Histograms

Pixels are counted column-wise and row-wise, thus two histograms:  $H_x$  and  $H_y$ , are created. Consider cumulative histograms  $V_x$  and  $V_y$ . Their  $k$ th bin expresses a total of first  $k$  bins of  $H_x$  and  $H_y$ , respectively [3]. By concatenating  $V_x$  and  $V_y$ , we get the feature vector, which is to some extent tolerant to shifting of glyph fragments.

### 2.4 Projection Axes

Image is fragmented into cells. Pixels present within each cell are cast orthogonally onto dedicated axes. In this approach features are identified with degrees of projection axes filling. The authors of [10] studied this technique coupled with Toeplitz model. Figure 3 depicts the cell system used by the authors.

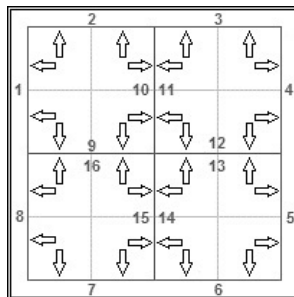


Fig. 3. A variant of projection axes technique

### 2.5 Central Moments

A moment is a scalar quantity that internally describes the shape of a function using powers of spatial variables. Potential application of moments in pattern recognition was first discovered by Hu and derives from the *uniqueness theorem* given in [11]. If we interpret an image as a pixel intensity function, then a set of moments becomes

shape descriptor. Translational invariance can easily be obtained by introducing central moments, which for discrete 2D function  $f(x, y)$  are expressed by (1):

$$\mu_{pq} = \sum_x \sum_y f(x, y)(x - \bar{x})^p(y - \bar{y})^q, \tag{1}$$

where  $p + q$  is the order of the moment and  $\{x, y\}$  is the centroid of the shape. Attention must be paid to the different orders of moment values magnitudes. This leads to unequal contribution from particular dimensions that hinders statistical classification. In order to compensate for this drawback, all components are multiplied by ten to the power of  $m - (p + q)$ , where  $m$  is the maximal order used (here: 5). The solution yields far better classification results than the logarithming and the normalization of features.

### 2.6 Hu Moments

From central moments Hu derived similitude invariants (2):

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\Gamma}, \tag{2}$$

where  $\Gamma = (p + q + 2) / 2$  and  $p + q > 1$ . On the basis of  $\eta_{pq}$  Hu constructed seven expressions invariant under general linear transformations (translation, scale and rotation), the final one also being invariant under skew [11].

Unfortunately, seven-dimensional descriptor may fail when paired with a statistical classifier. This is due to the incommensurability between vector elements caused, similarly to what was pointed out in Section 2.5, by varying and hardly predictable orders of magnitude. As a remedy, the authors multiply each invariant by arbitrarily chosen factor  $10^n$ , where  $n = 0, 1, 1, 1, 2, 2, 3$ , respectively.

### 2.7 Zernike Moments

The Zernike moment of order  $n$  and repetition  $m$  ( $A_{nm}$ ) is given by the inner product of a function  $f(x, y)$  and the Zernike polynomial  $V_{nm}(x, y)$ . For images the expression unfolds as in (3):

$$A_{nm} = \frac{n + 1}{\pi} \sum_x \sum_y f(x, y)V_{nm}^*(x, y), \tag{3}$$

$$x^2 + y^2 = \rho^2 \leq 1,$$

with “\*” to denote complex conjugate. For purposes of this definition, we assume that images are of unitary size. The confinement to the unit disk is a consequence of the definition of Zernike polynomials, which are a set of two-dimensional, orthogonal, complex functions. The properties and applications of Zernike moments were broadly investigated in [12].

Zernike moments owe their role in pattern recognition to two properties. First, their magnitudes are invariant to rotation. Second, the orthogonality of  $V_{nm}$  basis enables

reconstruction of an image from a set of moments by summing up consecutive image contributions (eq. 14 in [12]).

### 2.8 Unitary Transforms

Unitary transforms are a class of linear transformations which are both orthogonal and invertible. One major example of a unitary transform is Discrete Fourier Transform (DFT), which is widely utilized in digital signal processing. Thanks to orthogonality, the signal can be represented as a finite series expansion without any information redundancy. Thus, one may extract desired frequencies whilst discarding the other. As the transform is invertible, one can subsequently return with the modified signal to the original domain by simple addition of terms. Main motivations to do so are signal filtering and data compression. Low-pass filtering is a method for eliminating the number of variables needed for successful description and identification. A procedure of shape coding requires transforming an  $N \times N$  image into  $N \times N$  frequency components and selecting only a limited number of transform coefficients from the low frequency part of spectrum to build the feature vector.

The other worth mentioning members of unitary transforms group are: Karhunen-Loève Transform (KLT), Discrete Cosine Transform (DCT) and Discrete Hadamard Transform (DHT). According to [13], KLT emerges as the ultimate in terms of compactness, but we are short of an efficient way to compute it. The authors of the comparison concluded that DCT most closely matches the performance of KLT and they also developed a fast DCT algorithm. The transform together with DFT and DHT are thus investigated in this work as a tool of shape description. Selected unitary transforms were previously considered in [14] as global features for recognition of online handwritten numerals and Tamil characters.

If we represent a finite, periodic set of  $N$  complex samples by  $f(n)$ , then, as a result of DFT, we obtain an equinumerous set of complex coefficients  $F(k)$ , given by (4):

$$F(k) = \sum_{n=0}^{N-1} f(n)e^{-i2\pi k \frac{n}{N}} . \tag{4}$$

Hadamard transform matrix of size  $2N \times 2N$  ( $\mathbf{H}_{2N}$ ) consists solely of positive and negative ones and is defined recursively by (5):

$$\mathbf{H}_1 = [1] , \quad \mathbf{H}_{2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{bmatrix} . \tag{5}$$

Discrete Cosine Transform  $G(k)$  of set  $g(n)$  is given by (6):

$$G(0) = \frac{\sqrt{2}}{N} \sum_{n=0}^{N-1} g(n) , \tag{6}$$

$$G(k) = \frac{2}{N} \sum_{n=0}^{N-1} g(n) \cos \left( \frac{(2n + 1)k\pi}{2N} \right) , \quad k = 1, 2, \dots, N - 1 .$$

## 2.9 Polyline Approximation

Freeman (1961) [15] proposed a method of encoding contours as a sequence of numbers, expressing relative segment positions. Pixels are labelled with digits 0-7, dependant on where the adjacent pixel is situated. Freeman chain codes effectively describe the shape of a curve with segments connected with each other at a multiple of  $45^\circ$  angle. The concept can be easily extended by dividing a contour into  $N$  fragments of equal length and connecting the division points with vectors  $\mathbf{v}_l$  ( $l = 1, \dots, N$ ). If the curve cannot be divided without a remainder, it remains open. Each  $\mathbf{v}_l$  phase  $\varphi_l$  contributes to the feature vector  $\mathbf{V} = [\varphi_1, \varphi_2, \dots, \varphi_N]$ , which represents the coarse shape of the outline. The angular distance  $d_\phi$  serves for comparing two feature vectors  $\mathbf{U}$  and  $\mathbf{V}$ :

$$d_\phi(\vec{U}, \vec{V}) = \sum_{l=1}^N d_l(\vec{U}, \vec{V}) = \sum_{l=1}^N \left( 1 - \left| \frac{|\Delta\phi_l| - \pi}{\pi} \right| \right), \quad (7)$$

where  $\Delta\phi_l$  denotes the phase difference between  $\mathbf{v}_l$  and  $\mathbf{u}_l$  in radians. It should be noted that the technique is appropriate for encoding single contours only, which proves inconvenient for representing glyphs that can occur with a diacritic.

## 2.10 Elliptic Fourier Descriptors

Description of closed contours via parametric equations was considered among others in [16]. As a result of expanding a parametric curve in Fourier series, one gets coefficients that strictly depend on size, orientation and selection of starting point on a curve [17]. The essence of the problem is to construct expressions relating the expansion terms which would be free of undesired information and thus could be considered as pure shape features.

Kuhl-Giardina elliptic descriptors [16] approximate a contour by superposition of harmonic phasors that encircle ellipses reflecting particular sine-cosine pairs of  $x$  and  $y$  projection expansions as in (8) and (9):

$$x(t) \approx \frac{a_0}{2} + \sum_{n=1}^N \left[ a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right) \right], \quad (8)$$

$$y(t) \approx \frac{c_0}{2} + \sum_{n=1}^N \left[ c_n \cos\left(\frac{2\pi nt}{T}\right) + d_n \sin\left(\frac{2\pi nt}{T}\right) \right], \quad (9)$$

where  $N$  is the highest order and  $T$  is the total contour length. Expansion coefficients  $a_n, b_n, c_n, d_n$  have to be normalized with respect to three factors: phase shift  $\theta$ , orientation  $\psi$  and the semi-major axis length  $E$  of the first harmonic ellipse. The final outcome is a set of invariant coefficients  $a_n^{**}, b_n^{**}, c_n^{**}, d_n^{**}$ , which can be expressed in matrix notation (10):

$$\begin{bmatrix} a_n^{**} & b_n^{**} \\ c_n^{**} & d_n^{**} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} a_n & b_n \\ c_n & d_n \end{bmatrix} \begin{bmatrix} \cos(n\theta) & -\sin(n\theta) \\ \sin(n\theta) & \cos(n\theta) \end{bmatrix}. \quad (10)$$

The authors of [18] applied Kuhl-Giardina Fourier descriptors in their system for identification of handwritten Arabic characters.

### 3 Matter of Study

The authors undertake implementation and testing of the methods presented in Section 2. The works were conducted using the Java programming language with the support of *Apache Commons Math* library (fast transform algorithms, Section 2.8) [19]. All issues were related to binary images – glyphs of triple form: solid, thinned and outlines. K3M algorithm was utilized for thinning [20]. The task was targeted at relative comparison of description techniques, highlighting their strengths and weaknesses and optimization of performance by adjustment of parameters.

The tests were carried out on own collection of 2460 typewritten glyphs including Latin letters and numerals. The dataset is divided into 33 font subsets. Ten of them comprise 52 Latin alphabets (both upper and lower case) and ten digits. The remaining 23 scripts contain additional 18 Polish-specific characters (A, Ć, E, Ł, Ń, Ó, Ś, Ź, Ż and a, ć, e, ł, ń, ó, ś, ź, ż). The selection of fonts provides broad diversity of styles which is responsible for high representativeness of test results. The following font styles are present: serif (e.g. *Times New Roman*), sans-serif (e.g. *Arial*), console (e.g. *Consolas*), decorative (e.g. *Comic Sans*) and exaggerated (e.g. *Elephant*).

From the broad spectrum of classification algorithms, the authors decided to employ the simple k-nearest neighbor classifier with Manhattan metric and  $k = 2$ . The authors' implementation increases  $k$  each time there is a tie and repeats the procedure. The classifier is not designed to return "not recognized" response.  $k = 2$  is also used together with angular metric (refer to Section 2.9).

$$x_i \rightarrow \frac{x_i - \mu}{\sigma}, \quad (11)$$

Feature vectors may be subject to standarization:

where  $\mu$  denotes mean of components and  $\sigma$  is their standard deviation. The goal of this operation is elimination of significant differences between components, which can reach orders of magnitude, while keeping relations between them.

#### 3.1 Recognition Rate

Main quality criterion of a feature extraction technique is the percentage of correct identification. The examination tool is leave-one-out cross-validation. The analysis of recognition rate based upon the whole collection of glyphs may be unreliable due to the problems emerging while trying to distinguish lower and upper case counterparts of letters: C, O, S, V, W, X, Z, Ć, Ó, Ś, Ź, Ż. Therefore, any attempts of doing so are discarded and the corresponding classes are merged into one class, i.e.,  $c \equiv C$ ,  $z \equiv Z$ , etc. Individual subcategories of the complete set are also examined: letters, lower case letters, upper case letters and digits. Polish-specific alphabets are excluded from tests of contour descriptors due to strong presence of diacritics.



### 3.2 Execution Speed

The efficiency of a method is measured by the average time it takes to extract feature set from a single character. The value does not encompass previous operations of thinning and size normalization. The additional information is mean classification time, correlated to the dimensionality of descriptor. Both quantities are given with a precision of 1 ms. Alternative indices are: the number of identifications per second and the time needed to process one A4 page with 30 lines of text, each containing 70 characters in average (2100 characters per page). The computations are carried out on processor *Intel Core 2 Duo T6600*, 2.2 GHz. Two factors which may negatively affect the speed are: object-oriented nature of the Java programming language and suboptimal quality of algorithms.

## 4 Experimental Results

In this section results of investigation categories described in Section 3 are given and discussed. Table 1 sets together the characteristics of particular descriptors. Image size, glyph form and the dimensionality of descriptors were adjusted with a view to achieving highest possible rates.

**Table 1.** Descriptor parameters. In column <Image form>: 'S' stands for 'solid', 'T' for 'thinned' and 'C' for 'contour'.

Technique	Descriptor constituents	No. of features	Image form	Image size	Vector standardized?
<b>Zoning</b>	pixel density in each zone, row and column	69	S	60×90	yes
<b>Crossings</b>	averaged coordinates of crossing points	20	S	63×63	yes
<b>Projection Histograms</b>	cumulative bin counts from both histograms	130	T	65×65	yes
<b>Projection Axes</b>	lengths of projections	16	S	64×64	yes
<b>Central Moments</b>	all central moments of orders 2-5	18	S	32×32	no
<b>Hu Moments</b>	seven Hu moment invariants	7	T	41×41	no
<b>Zernike Moments</b>	all Zernike moments of orders 2-8 and $m \geq 0$	23	T	48×48	no
<b>Fourier Transform</b>	a number of low-frequency samples	224	S	32×32	yes
<b>Hadamard Transform</b>		416	S	32×32	yes
<b>Cosine Transform</b>		320	S	32×32	yes
<b>Polyline Approximation</b>	polyline segment phases	12	C	any	no
<b>Elliptic Descriptors</b>	seven first invariant series coefficients	25	C	any	no

#### 4.1 Results of Classification

Table 2 shows the percentages of correct classifications for each feature extraction method with regard to particular glyph class. The examination makes it obvious that pixel distribution methods have clear edge over other classes in terms of recognition rate of glyphs with known orientation. The crossings technique seems to be the method of choice. Cosine Transform and Zernike moments are not far behind despite the latter inability to correctly recognize pairs like 6 and 9. The rotation invariance is thus partially responsible for classification errors. Contour description methods are rather uncompetitive. Low rates can be explained by awkwardness of closed contour labelling algorithm, which is no silver bullet against many ambiguities and nontypical cases emerging while encoding chain of curve points. The least promising description technique is the one based on Hu moments, which suffers from low dimensionality.

**Table 2.** Classification results

Technique	All glyphs	Letters	Lower case letters	Upper case letters	Digits
<b>Zoning</b>	89.8%	91.9%	95.4%	93.3%	97.0%
<b>Crossings</b>	90.9%	93.5%	95.6%	95.5%	95.8%
<b>Projection Histograms</b>	90.9%	93.1%	94.3%	92.9%	93.6%
<b>Projection Axes</b>	89.7%	92.3%	94.6%	92.7%	96.7%
<b>Central Moments</b>	81.5%	84.5%	90.1%	85.3%	91.8%
<b>Hu Moments</b>	47.0%	n/a	n/a	n/a	n/a
<b>Zernike Moments</b>	86.5%	89.2%	89.0%	93.5%	92.4%
<b>Fourier Transform</b>	76.5%	79.5%	81.7%	84.1%	81.8%
<b>Hadamard Transform</b>	78.1%	79.7%	80.9%	80.5%	90.3%
<b>Cosine Transform</b>	87.2%	88.8%	91.4%	88.7%	95.8%
<b>Polyline Approximation</b>	78.5%	79.5%	82.6%	83.0%	89.4%
<b>Elliptic Descriptors</b>	75.7%	78.1%	80.5%	80.7%	78.2%

#### 4.2 Time Efficiency

The superiority of zoning, crossings and histograms was reassured by average identification rates not exceeding 4 ms per glyph. Descriptors based on unitary transforms owe their agility to well-known fast processing algorithms. Regretfully, because of their high dimensionality, the techniques are slightly delayed by lengthy classification process. Previously asserted high accuracy of projection axes is severely hindered by relatively low time efficiency. As before, contour-based descriptors heavily suffer from drawbacks of chain encoding algorithm. Extreme complexity of Hu invariants is again reflected by unacceptable extraction rates. It cannot be negated that the benefits brought by geometric invariance may not compensate for the price one has to pay for them. Table 3 aggregates average identification rates of a single glyph together with the alternative quantities to better imagine the potential of particular techniques.

**Table 3.** Time efficiency rates

Technique	Extraction rate	Classification rate	No. of identifications per 1 sec	Time of one A4 page analysis
<b>Zoning</b>	1 ms	2 ms	333	6.3 s
<b>Crossings</b>	2 ms	2 ms	250	8.4 s
<b>Projection Histograms</b>	1 ms	3 ms	250	8.4 s
<b>Projection Axes</b>	12 ms	2 ms	71	29.6 s
<b>Central Moments</b>	14 ms	2 ms	62	33.6 s
<b>Hu Moments</b>	75 ms	n/a	13	157.5 s
<b>Zernike Moments</b>	17 ms	2 ms	52	39.9 s
<b>Fourier Transform</b>	< 1 ms	5 ms	200	10.5 s
<b>Hadamard Transform</b>	< 1 ms	7 ms	142	14.7 s
<b>Cosine Transform</b>	< 1 ms	5 ms	200	10.5 s
<b>Polyline Approximation</b>	31 ms	3 ms	29	71.4 s
<b>Elliptic Descriptors</b>	32 ms	1 ms	30	69.3 s

## 5 Conclusion

Tests conducted in the presented work identified contenders for the most useful feature extraction method for OCR purposes with application to machine-printed Latin text. Pixel distribution techniques boast the highest number of successful classifications and the time efficiency rates. Recognition rate peaks of 90% may not impress in absolute terms, nevertheless one should consider the number of glyph classes and the high diversity of font styles. The DCT descriptor emerged as a very close competitor, also thanks to the application of fast algorithm. Zernike moments also seem to be successful, albeit their rotational invariance may prove a curse. The computational expense of the technique is also not very appealing. The authors judge that contour description techniques should perform better in combination with real object outlines.

**Acknowledgment.** The research was supported by the Rector of Bialystok University of Technology in Bialystok, grant number S/WI/1/2013.

## References

1. Sazaklis, G.N.: Geometric Methods for Optical Character Recognition. PhD dissertation. State University of New York at Stony Brook (1997)
2. Devijver, P.A., Kittler, J.: Pattern Recognition: a Statistical Approach, p. 12. Prentice-Hall, London (1982)
3. Trier, O.D., Jain, A.K., Taxt, T.: Feature Extraction Methods for Character Recognition – a Survey. *Pattern Recognition* **29**(4), 641–662 (1996)
4. Lorigo, L.M., Govindaraju, V.: Offline Arabic Handwriting Recognition: a Survey. *IEEE Trans. on Pat. Anal. and Mach. Int.* **28**(5), 712–724 (2006)

5. Jayadevan, R., Kolhe, S.R., Patil, P.M., Pal, U.: Offline Recognition of Devanagari Script: a Survey. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews* **41**(6), 782–796 (2011)
6. Eikvil, L.: OCR – Optical Character Recognition. Norsk Regnesentral. Report No. 876 (1993)
7. Saeed, K., Tabezki, M.: Cursive-character script recognition using toeplitz model and neural networks. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) *ICAISC 2004. LNCS (LNAI)*, vol. 3070, pp. 658–663. Springer, Heidelberg (2004)
8. Saeed, K.: Carathéodory-Toeplitz based mathematical methods and their algorithmic applications in biometric image processing. *Applied Numerical Mathematics* **75**, 2–21 (2014). Elsevier
9. Pradeep, J., Srinivasan, E., Himavathi, S.: Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network. *IJCSIT* **3**(1), 27–38 (2011)
10. Saeed, K.: A projection approach for arabic handwritten characters recognition. In: Sincak, P., Vascak, J. (eds.) *Quo Vadis Computational Intelligence? New Trends and Applications in Computer Intelligence*, pp. 106–111. Physica-Verlag, Berlin - Kacprzyk J. (Ed) (2000)
11. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory* **IT-8**, 179–187 (1962)
12. Khotanzad, A., Hong, Y.H.: Invariant Image Recognition by Zernike Moments. *IEEE Trans. on Pat. Anal. and Mach. Int.* **12**(5), 489–497 (1990)
13. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete Cosine Transform. *IEEE TC* **C-23**(1), 90–93 (1974)
14. Ramakrishnan, A.G., Urala, K.B.: Global and local features for recognition of online handwritten numerals and tamil characters. In: *MOCR 2013 Proceedings of the 4th International Workshop on Multilingual OCR* (2013)
15. Freeman, H.: On the Encoding of Arbitrary Geometric Configurations. *IRE Trans. Elec. Comp.* **EC-10**, 260–268 (1961)
16. Kuhl, F.P., Giardina, C.R.: Elliptic Fourier Features of a Closed Contour. *Computer Graphics and Image Processing* **18**, 236–258 (1982)
17. Granlund, G.H.: Fourier Preprocessing for Hand Print Character Recognition. *IEEE TC* **C-21**(2), 195–201 (1972)
18. Maddouri, S.S., Amiri, H., Belaid, A., Choisy, Ch.: Combination of local and global vision modelling for arabic handwritten words recognition. In: *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, pp. 128–135 (2002)
19. Commons Math: The Apache Commons Mathematics Library, ver. 3.3, June 22, 2015. <http://commons.apache.org/proper/commons-math/>
20. Saeed, K., Tabezki, M., Rybnik, M., Adamski, M.: K3M: A Universal Algorithm for Image Skeletonization and a Review of Thinning Techniques. *AMCS* **20**(2), 317–335 (2010)