Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-75

On handwriting preprocessing for 2D object recognition systems

[Marcin Woźniak, Dawid Połap, Zbigniew Marszałek]

***Abstract*— This paper illustrates object preprocessing using some interpolation methods. Interpolation is used to obtain the best angle of inclination of input curve based on human signature. This feature is one of crucial aspects for efficiency in object recognition systems that are basing on handwriting signature. In the article is presented analysis of selected methods that may help in preprocessing of handwriting for artificial intelligence systems.**

Keywords*—*handwriting recognition, 2D object interpolation.

I. Introduction

Handwriting recognition is classified as one of basic topics in pattern recognition in artificial intelligence (AI). Before using AI methods to classify given pattern one need to focus on proper representation of input signature. To identify someone, we may focus on handwriting, especially signature. Some interesting recognition methods are presented in [2,4,17,21]. During lifetime, handwriting may differ along with age or changes in type of writing, size or topography of letters and many other circumstances. These features make it peculiar and unique for each person. Therefore it is possible to determine if text was or not written by someone. It may not only be used in graphology but is also widely used in identification. For example, handwritten signature is a certain way to counter check identity of respondent in bank or any other institution. The main problem is to find best way to digitalize input signature. In digitalization we must properly find edges of transformed object. Some ideas to digitalize handwriting are discussed in [6,16,20,22]. The other problem is reforming transformed object into right shape for AI classifiers, see [12,17]. Nowadays, signature is used in many devices: in banks to confirm customers personality, at work to sign documents or confirm identity.

Verification of the signature in electronic device goes along with some schema. User is signing in provided area. Then signature is transformed into electronic version which can be recognized by dedicated tool. In this process, one need to identify key points in the object we are digitalizing or transforming for AI systems. Some interesting methods of shape matching and interpolation are presented in [2,3,9]. In [5,10] are presented examples of key point finding and image classification. This makes it important to have proper and valid systems of handwriting recognition. There are many ways of recognition, see [4,12,14,17,18,19]. This problem becomes even more complicated in 3D, as shown in [1,13,14]. It is complex problem, which has not been entirely investigated to this day. One of reasons for such situation is different handwriting style. Without doubt, everyone has peculiar style of writing. But suppose, a person who signs document will also twitch or for some reasons cannot do it correct. The signature is deformed, however we may not ask to sign again. Another problem is location or rotation of the signature in designated area. This happens very often, for example if we sign something using electronic devices with provided area of writing. For example in a hurry, client will write signature along with any edge of provided area. In this case, there is need to rotate the signature and then prepare it for validation. Solution to this problem can be sought in various ways. Mainly we define it as problem of data form and the method to collect it (data acquisition).

Signature is 2D object. Moreover we can treat it as curve in the surface. Therefore one may apply typical mathematical operations to form it in the way our AI classifier may need. If we treat a signature as curve we may interpolate it using key points. Interpolation is one of important mathematical applications in AI systems. In [4,15,16,21] are presented it's applications in filtering or processing irregularly spaced data. Interpolation will be very helpful in situations when user's signature is deformed, rotated or we simply need to relocate it or resize it. This topic is investigated in presented examinations.

II. Preprocessing 2D input object

There are primarily two data acquisition methods: passive and dynamic. Passive data acquisition means collecting data from simulation. Dynamic data acquisition means collecting data directly from the user. This is very important problem in handwriting recognition. Input for AI pattern recognition systems should have a common format, for instance binary. For this type we have to prepare the object. There is a problem of provided area, because we use different types of electronic devices. However we want user to write freely and make

Marcin Woźniak
Institute of Mathematics, Silesian University of Technology
Gliwice, Poland

Dawid Połap
Institute of Mathematics, Silesian University of Technology
Gliwice, Poland

Zbigniew Marszałek
Institute of Mathematics, Silesian University of Technology
Gliwice, Poland

computer recognize the object, here respondent signature, properly.

We have investigated situations when computer shall find signature placed anywhere in provided area and then if needed cut it, rotate or resize to have best possible input data for AI recognition system. In our research was used 530x270 pixels touch pad, provided for the users to sign. The system was transforming input signature from 530x270 pixels object into 40x15 pixels object. To solve it we have used interpolation on selected key points. In the article, we have examined some selected ways of interpolating to verify the best one for handwriting (signature) AI recognition systems. In [4,13,15,16,21] we can find detailed description of interpolation methods. While in [18,19] are described some other interesting methods to preprocess input data for AI recognition systems. We can define interpolation as transformation from input curve to resized one, but without changing most important facts. This makes transformed object similar to use for pattern recognition. The research sample is handwritten signature. Many devices use different input appliance, single or sophisticated resolution. Also users may use different language or have different handwriting type. Therefore it is essential to interpolate all possible input elements in a special, predefined way. Human signature, from mathematical point of view, is continuous or broken line, and that is why we can treat it as curve. Therefore, it is important to examine which of interpolation method is most efficient. Let us now discuss some sample mathematical methods.

A. *Linear interpolation*

First examined method is linear interpolation. To make an object linearly interpolated we use polynomial function

$$f(x) = a_0 + a_1 x\,, \tag{1}$$

where a_0 and a_1 are constants that we are searching for. Assuming that we have *n* input points (x_i, y_i) for $i = 1, \ldots, n$ we are looking for polynomial which fulfills condition $f(x_i) = y_i$ for $i = 1, \ldots, n-1$. Thus interpolation has the following form

$$S(a_0, a_1) = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2\,. \tag{2}$$

Therefore system of normal equations is

$$\begin{cases} \dfrac{\partial S(a_0, a_1)}{\partial a_0} = \sum\limits_{i=1}^{n} (y_i - a_0 - a_1 x_i)(-1) = 0 \\ \dfrac{\partial S(a_0, a_1)}{\partial a_0} = \sum\limits_{i=1}^{n} (y_i - a_0 - a_1 x_i)(-x_i) = 0 \end{cases}. \tag{3}$$

After basic mathematical operations we have

$$a_0 n + a_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i\,, \tag{4}$$

$$a_0 \sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i\,, \tag{5}$$

However in numerical experiments one may use matrices *A*, *X*, *Y* defined as system of normal equations and it's coefficients. Therefore linear normalized function in (2), may be presented as

$$A = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}, Y = \begin{bmatrix} \sum\limits_{i=1}^{n} y_i \\ \sum\limits_{i=1}^{n} x_i y_i \end{bmatrix}, X = \begin{bmatrix} \sum\limits_{i=1}^{n} 1 & \sum\limits_{i=1}^{n} x_i \\ \sum\limits_{i=1}^{n} x_i & \sum\limits_{i=1}^{n} x_i^2 \end{bmatrix}. \tag{6}$$

To find coefficients we use matrix equation $A = X^{-1} Y$. One may also use quadratic or higher polynomials to interpolate input signature. This will be similar in form to (6).

B. *Quadratic interpolation*

Quadratic interpolation is similar to method in section II.A, but in this case our function is

$$f(x) = a_0 + a_1 x + a_2 x^2\,, \tag{7}$$

where a_0, a_1 and a_2 are constants. Function $S(a_0, a_1, a_2)$ for points (x_i, y_i) for $i = 1, \ldots, n$ is

$$S(a_0, a_1, a_2) = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2\,. \tag{8}$$

System of equations has the following form

$$\begin{cases} \dfrac{\partial S(a_0, a_1, a_2)}{\partial a_0} = \sum\limits_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)(-1) = 0 \\ \dfrac{\partial S(a_0, a_1, a_2)}{\partial a_1} = \sum\limits_{i=1}^{n} (y_i - a_0 - a_1 x_i)(-x_i) = 0 \\ \dfrac{\partial S(a_0, a_1, a_2)}{\partial a_2} = \sum\limits_{i=1}^{n} (y_i - a_0 - a_1 x_i)(-x_i^2) = 0 \end{cases}. \tag{9}$$

After ordering we obtain

$$a_0 n + a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i\,. \tag{10}$$

When one use polynomial of higher order, there are many methods to search for interpolating curve. We chose one of them, where the matrices may be omitted because Cramer's formulas are sometimes numerically inefficient in calculations.

C. *Lagrange interpolation*

Cramer's formulas show that in polynomial $W_n(x) = a_0 x^0 + a_1 x^1 + \ldots + a_n x^n$ values $y_0, y_1, \ldots, y_n$ are only in first power. This interpolation is similar to previous methods, but here our basic function is

$$W_n(x) = y_0 \Phi_0(x) + y_1 \Phi_1(x) + \ldots + y_n \Phi_n(x)\,, \qquad (11)$$

where polynomial $W_n(x)$ is n degree and polynomials $\Phi_0(x), \ldots, \Phi_n(x)$ are formed as result of grouping. Degree of $\Phi_i(x)$ is less or equal to *n*, where $i = 1, \ldots, n$. We also assume that $W_n(x_i) = y_i$ for $i = 1, \ldots, n$ and any $y_0, y_1, \ldots, y_n$. From that we have $\Phi_i(x_j) = \delta_{ij}$ and $\Phi_i(x_j) = i$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$. Where $\Phi_i(x)$ resets this polynomial in all nodes of interpolation. Therefore to find Lagrange polynomials one may use formula

$$L_i(x) = \frac{\prod_{j=1, j\neq i}^{n} (x - x_i)}{\prod_{j=1, j\neq i}^{n} (x_i - x_j)}\,. \qquad (12)$$

D. *Chebyschev interpolation*

Last method, which was examined is Chebyshev interpolation. In this process we use Chebyshev polynomials of the first kind, recursively created using general equation

$$T_{n+1}(x) = 2x \cdot T_n(x) - T_{n-1}(x)\,. \qquad (13)$$

In numerical experiments were only used first five polynomials, which are: $T_0(x) = 1$, $T_1(x) = x$, $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$ and $T_4(x) = 8x^4 - 8x^2 + 1$. Therefore interpolating function $f(x)$ is

$$f_m(x) \approx \frac{1}{2} c_0 + \sum_{j=1}^{m} c_j T_j(x_k)\,. \qquad (14)$$

For discrete input, coefficients c_j are calculated with formula

$$c_j = \frac{2}{m+1} \sum_{k=0}^{m} f(x_k) T_j(x_k)\,. \qquad (15)$$

Let us now present research results for different kinds of input signature.

III. Research results

In numerical experiments were examined some examples of signatures. Input objects were rotated, twitched or multi line signatures, which may come in real situations. Input signatures were simplified by computer to create 2D curve without any doubled points in it. In [5, 10] are presented some efficient methods to find key points in pictures, which in simplified versions may be also applied to signatures, as in this paper. Simplified object, in the research, is therefore simplified curve containing only most important points. In case, when for one value of *y* were more arguments *x*, program calculated arithmetic average of *x* values. This assumption helps to simplify input object, even if it is complicated structure in 2D. Input objects after simplification according to presented rules were interpolated. Interpolation is to make acquisition of data (here handwritten signature) for AI pattern recognition system. Interpolation process was to transform input signature from 530x270 pixels object (touch pad input) into 40x15 pixels object (AI classifier).

Examples of signatures in original shape as written by the user on touch pad input, simplified versions and interpolated ones for AI classifier are presented in figures 1-24.

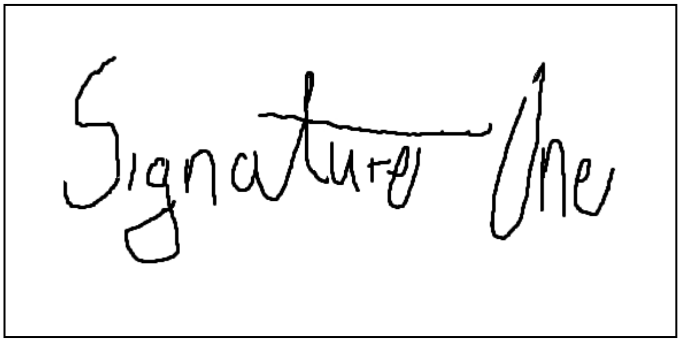

Figure 1. First sample in original shape as written by the user.

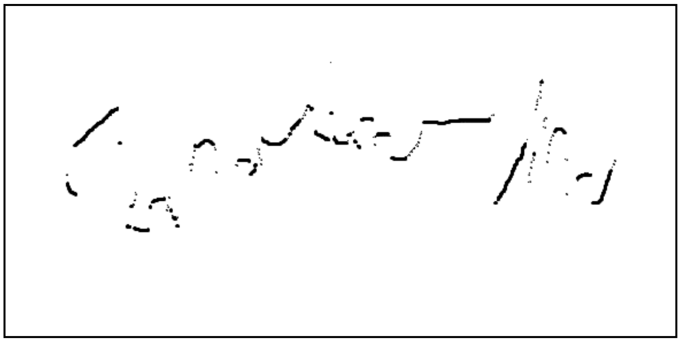

Figure 2. Simplification of first input object.

Curves of the first input signature are interpolated using methods presented in sections II.A – II.D. Interpolation is presented in figure 3. In the research computer treats it as a stream of values, representing simplified and interpolated input signature. Interpolated object is prepared for AI classifier. Some examples of data acquisition or systems

Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-75

positioning for AI systems by the use of evolutionary computation are presented in [7,8,11].

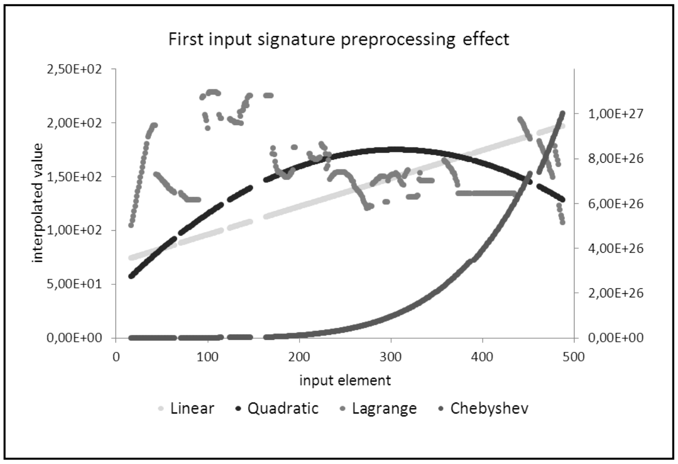

Figure 3. Interpolation of first 2D input object.

In figure 3, we see that all interpolation methods create almost continuous line. However Lagrange interpolation is separated and do not have compact shape.

Let us now discuss other examined signatures. We have analyzed twitched signatures or blurred letters on the input. We also analyzed rotated objects, as this is common situation in reality. These situations happen often if we write in hurry or there is soft ground or poor writing instrument. Therefore analysis of some examples is important. In signature two on the input, there is rotated object, as presented in figure 4.

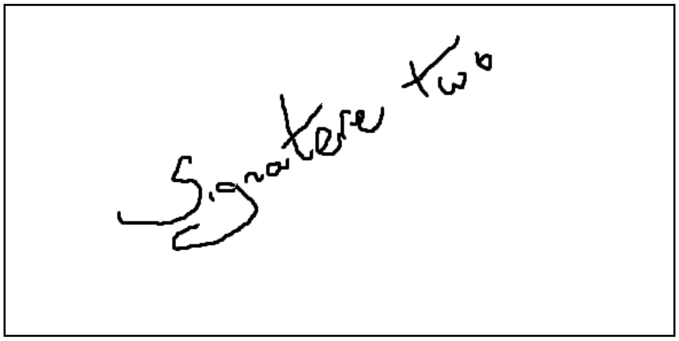

Figure 4. Second sample in original shape as written by the user.

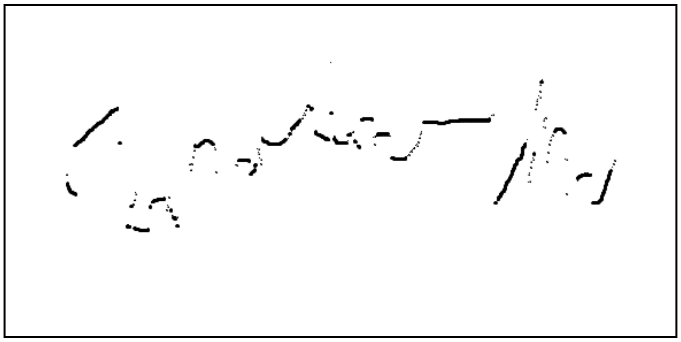

Figure 5. Simplification of second input object.

In figure 5 are presented key points of the first input signature. Because input letters were clear and not overlapping, simplified version contains many points.

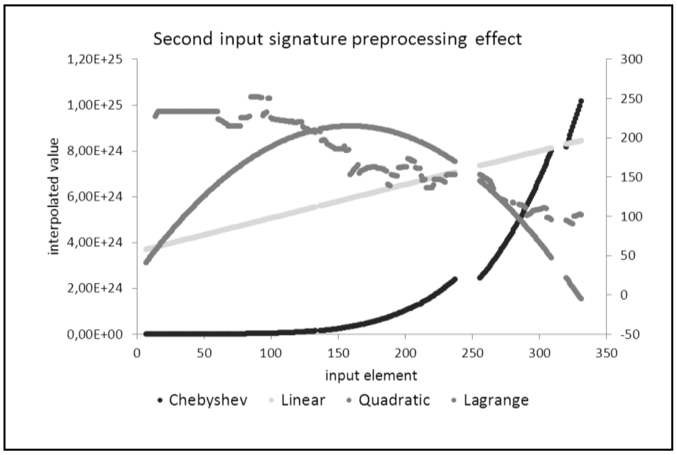

Figure 6. Interpolation of second 2D input object.

In figure 6 is presented interpolation of second input signature. We present all results of interpolation for discussed in section II methods in one chart for each input object. Input letters were clear without blurring or overlapping and interpolation effects are almost continuous lines and Lagrange points without big fluctuations.

Let us see third signature on the input, there are blurred, illegible and twitched letters, as presented in figure 7.

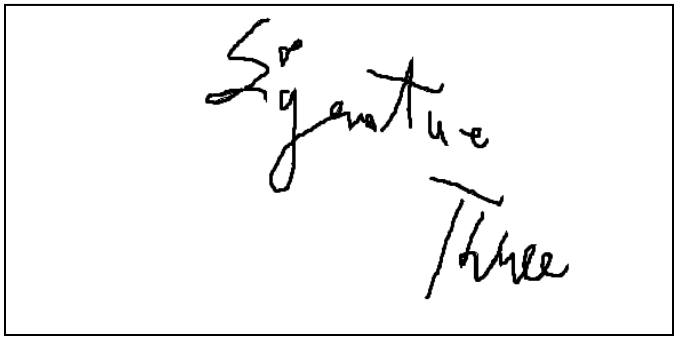

Figure 7. Third sample in original shape as written by the user.

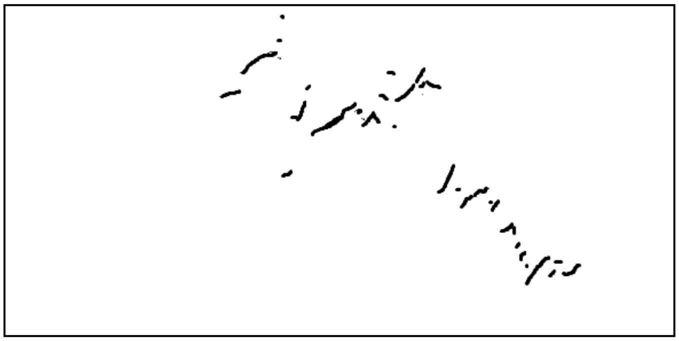

Figure 8. Simplification of third input object.

Simplified object is presented in figure 8. Blurred and illegible letters made it more complicated to find key points.

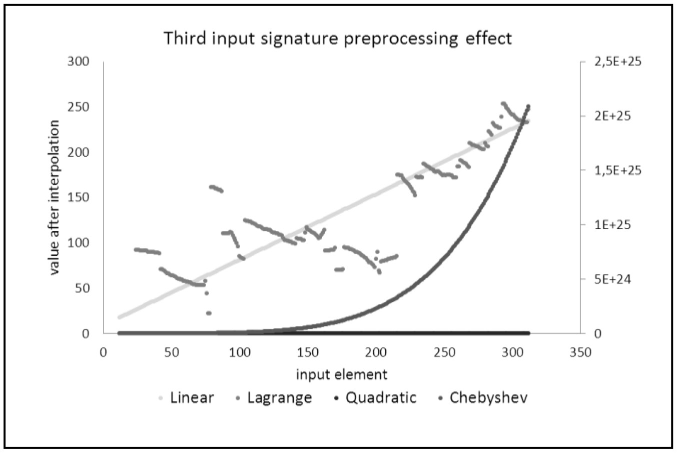

Figure 9. Interpolation of third 2D input object.

In figure 9, we see that blurred and illegible letters influenced interpolation effect.

Let us now discuss fourth input object. In signature four, on the input we have big and clear letters. However some letters are overlapping, as presented in figure 10.

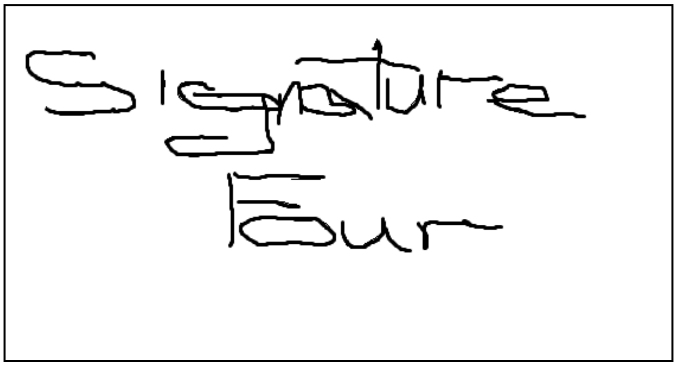

Figure 10. Four sample in original shape as written by the user.

Input object was simplified to get key points, as presented in figure 11.

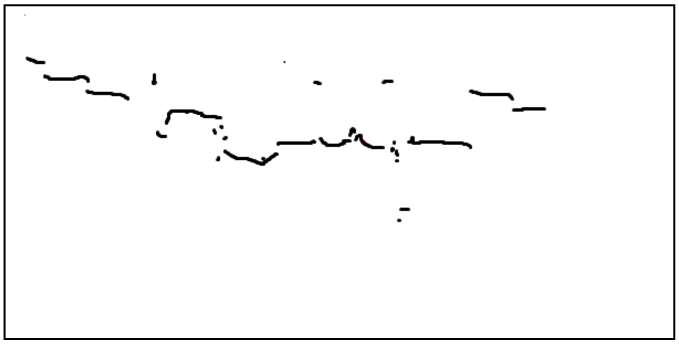

Figure 11. Simplification of four input object.

In figure 11 is presented simplified input signature. There was no rotation on the input, therefore key points are also horizontally placed.

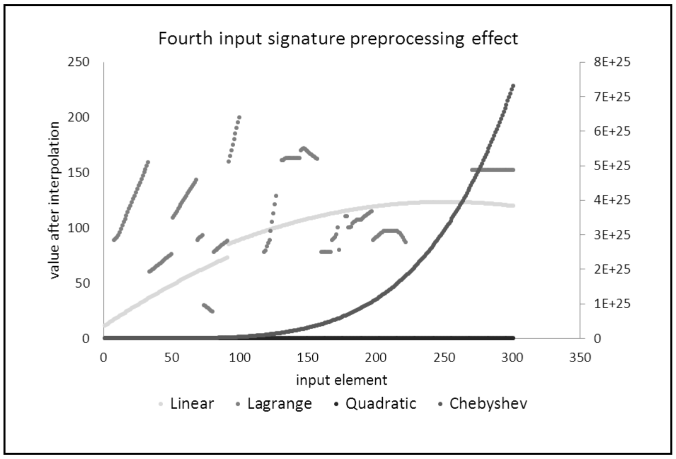

Figure 12. Interpolation of fourth 2D input object.

In figure 12, we see that big letters on the input affected compactness of interpolation. We see big fluctuations in Lagrange points.

Let us now present fifth input object. In signature fifth, there are clear letters. All input object is readable, as presented in figure 13.

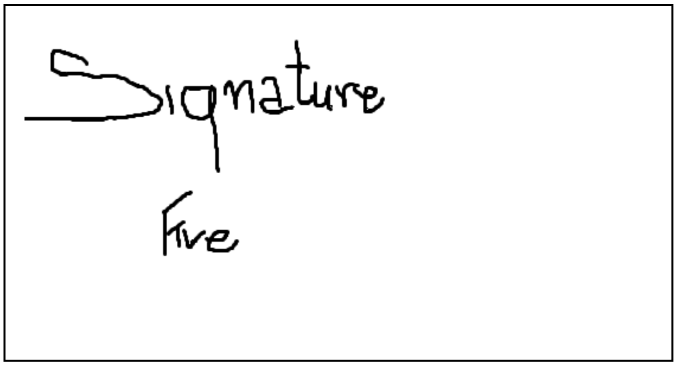

Figure 13. Fifth sample in original shape as written by the user.

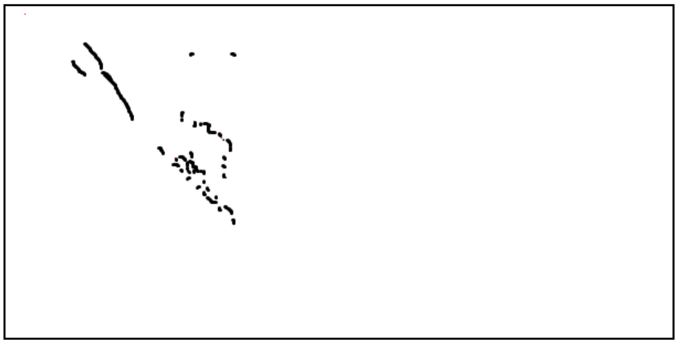

Figure 14. Simplification of fifth input object.

Simplified object is presented in figure 14. Object is clear, however simplified version looks like rotated, therefore in interpolation we can see fluctuations in Lagrange points.

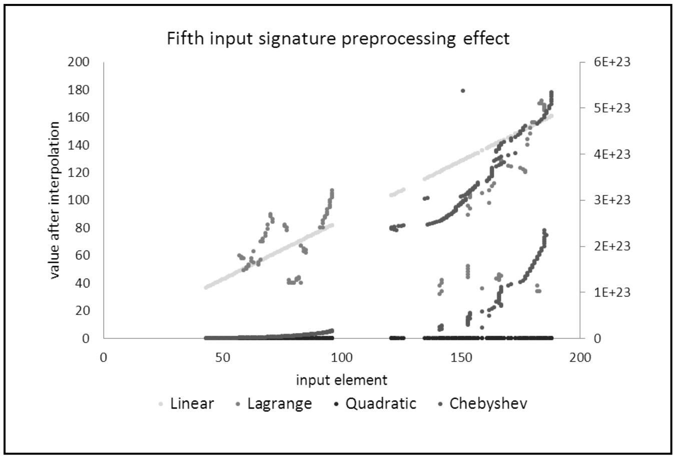

Figure 15. Interpolation of fifth 2D input object.

In sixth signature on the input, there are blurred, illegible and twitched letters. Moreover letters are also overlapping and the input signature is rotated, as presented in figure 16.

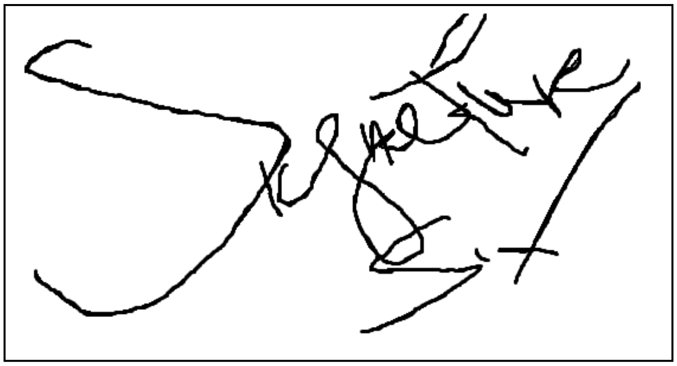

Figure 16. Sixth sample in original shape as written by the user.

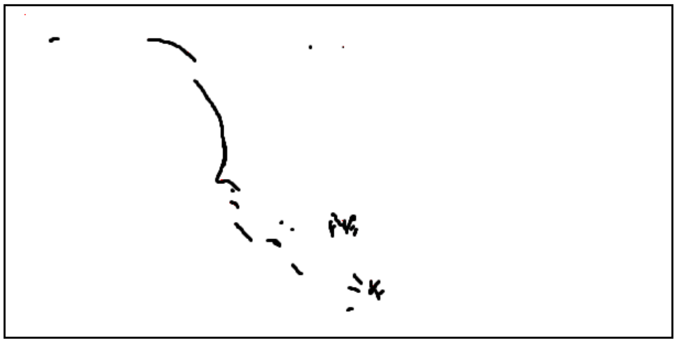

Figure 17. Simplification of sixth input object.

Simplified version of sixth input is presented in figure 17. Input signature was blurred, twitched and rotated, therefore key points are reflecting nature of this signature.

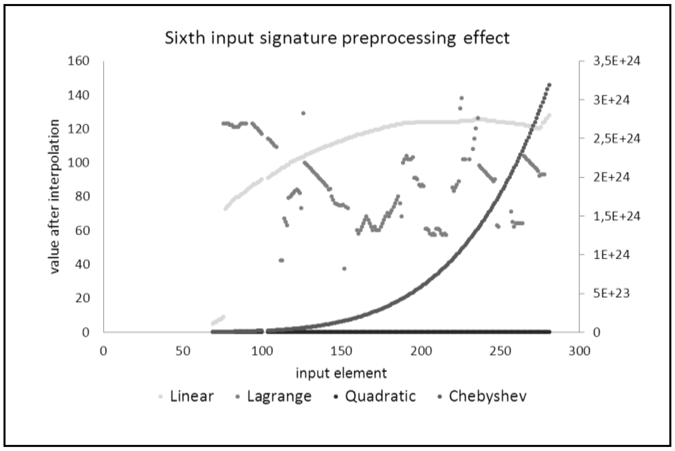

Figure 18. Interpolation of sixth 2D input object.

Nature of sixth input signature is also visible in interpolation effects. Lagrange points are fluctuated and other methods gave not compact lines.

In seventh signature on the input, words are rotated but letters are clear without blurring. Input object is not illegible or twitched, as presented in figure 19.

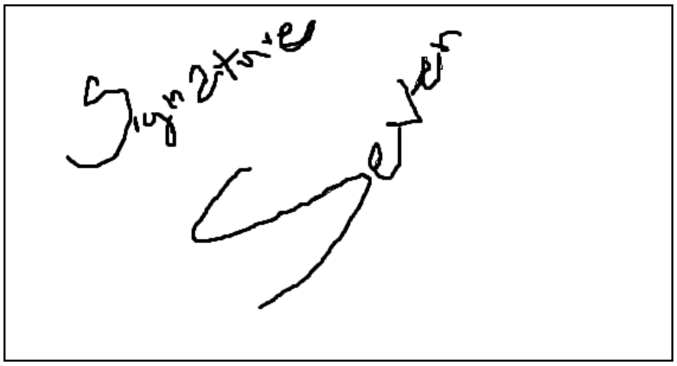

Figure 19. Seventh sample in original shape as written by the user.

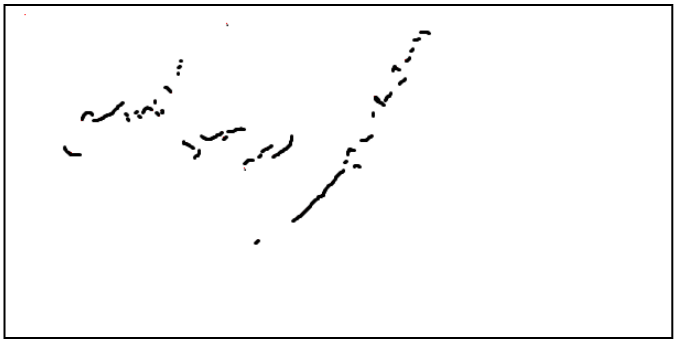

Figure 20. Simplification of seventh input object.

Simplified object is presented in figure 20. We can see that rotation of the input object is important for simplification. In interpolation, presented in figure 21 we can see that there are

Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-75

fluctuations in Lagrange points. Other methods gave compact lines, which are all placed in similar area.

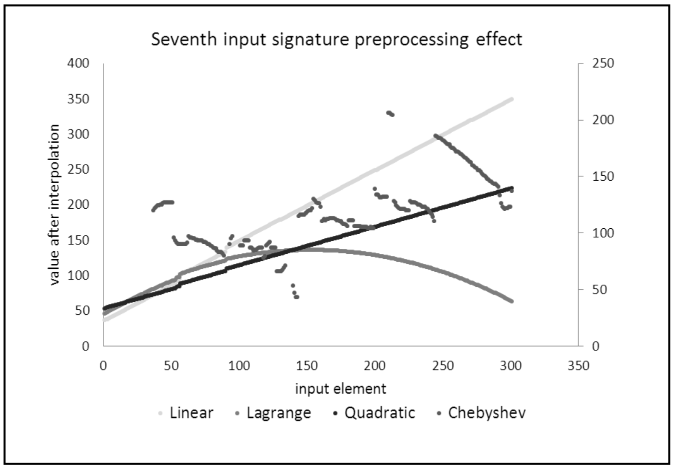

Figure 21. Interpolation of seventh 2D input object.

In eighth signature on the input, there are blurred, illegible and twitched letters. Moreover, letters are also overlapping and the input signature is rotated, as presented in figure 22.

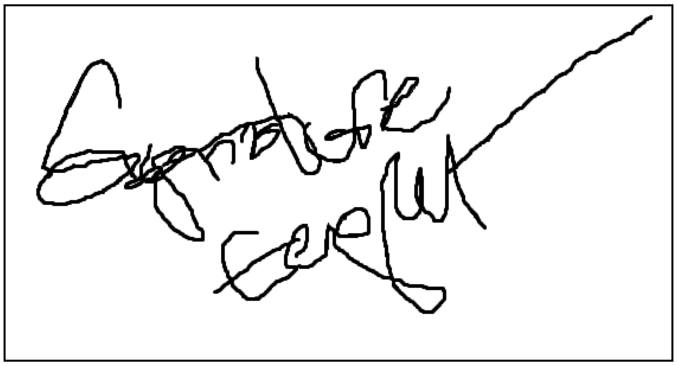

Figure 22. Eighth sample in original shape as written by the user.

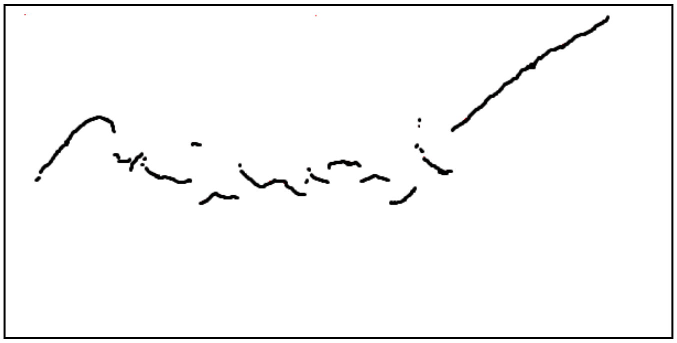

Figure 23. Simplification of eighth input object.

Analyzing key points after simplification we see that this input was hard to simplified. Interpolation of the input objects is presented in figure 24. One can notice that Lagrange points are fluctuated and other interpolations are placed rather vertically in the area.

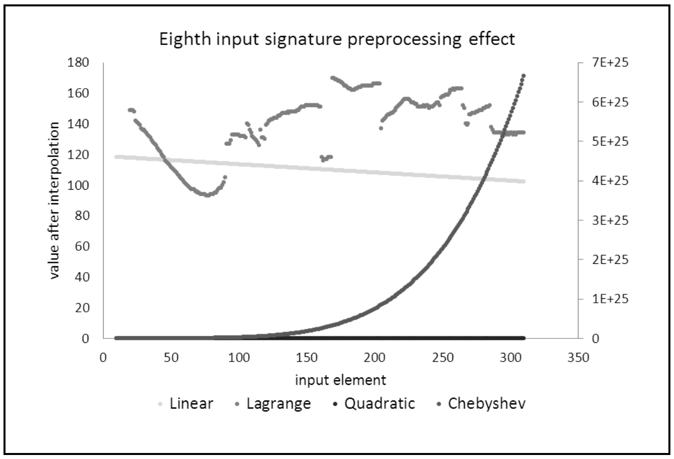

Figure 24. Interpolation of eighth 2D input object.

One can notice that even twitched or rotated object can be interpolated into size of AI classifier may need. It is also possible that preprocessing in some cases may interpolate less important key points.

A. *Comparison and assessment*

In preprocessing were used selected interpolation methods. Therefore it is important to measure accuracy and possible error. However for all presented examples the Runge effect may exists. One may describe it as deteriorating effect of polynomial interpolation despite increasing number of nodes (interpolation key points). This effect occurs if we use discontinuous function and if distances between all nodes are of equal length. Node in the research, we understand it as key point of simplified input signature. In our research, Runge effect unfortunately exists and it is visible. Absolute error is defined in formula

$$\varepsilon = \sqrt{\sum_{i=1}^{n} (p(x_i) - f_i)^2} \,, \qquad (16)$$

which best estimates an error for single value. Many peculiar points in 2D space we assess using formula

$$\varepsilon = \left| 0 - f(x_0) \right| , \qquad (17)$$

where ε is error value we measured and $f(x_0)$ is value of interpolation at measured point.

Let us now present accuracy of preprocessing for examined 2D input objects. In table 1 is presented accuracy for each input signature and average value. This represents expected accuracy of interpolation for given input signature.

TABLE I. ACCURARCY OF INTERPOLATION

Signature	**Interpolation type**			
	Linear	***Quadratic***	***Lagrange***	***Chebyschev***
1	56%	52%	58%	60%
2	58%	59%	54%	61%
3	56%	55%	50%	55%
4	59%	60%	41%	59%
5	65%	65%	28%	65%
6	49%	49%	31%	49%
7	72%	72%	53%	73%
8	70%	70%	69%	70%
Avg.	61%	60%	48%	62%

Accuracy of examined interpolation methods

Analyzing research results one may see that for common signatures without any special disturbances Quadratic or Chebyshev methods can give best results. Moreover the same methods work also very good with rotated or displaced input objects, what is visible in better accuracy. Worst results were achieved for twitched, blurred or overlapping signatures, where all examined methods work with similar accuracy but not Lagrange method. Therefore it seems to be necessary to work on methods for unclear input objects, that in practice may occur often.

IV. Conclusions

In our research, we have considered four different interpolation methods for preprocessing 2D input objects. Result show that to interpolate curve, which is based on human signature, one may use *n* degree polynomials. For instance: linear function (1st degree polynomial), quadratic function (2nd degree polynomial) or some more sophisticated. Depending on input data, preprocessing may result in some errors, but we must remember that input line is not always continuous and thus good interpolation nodes (input object key points) are very important. From the research we conclude that Lagrange method may cause low accuracy in some cases. By comparison, Chebyshev interpolation transferred more points, but the error is still high. Good results may give linear or quadratic method, both achieved similar results. It also seems that implementing new methods based on genetic or heuristic attempt to track and interpolate user's input signature using key points of preprocessed object may help to solve problem of rotated, twitched, blurred or overlapping signatures.

References

[1] T. D. Alter, and D. W. Jacobs, “Error Propagation in Full 3D-from-2D Object Recognition”, Computer Vision and Pattern Recognition, IEEE, 1994, pp. 892-896.

[2] S. Belongie, J. Malik and J. Puzicha, “Shape Matching and Object Recognition Using Shape Contexts”, Trans. on Pattern Analysis and Machine Intelligence, IEEE, 2002, vol. 24, no. 24, pp. 509-522.

[3] I. Bloch, “Fuzzy Relative Position Between Objects in Image Processing: A Morphological Approach”, Trans. on Pattern Analysis and Machine Intelligence, IEEE, vol. 21, no. 7, 1999, pp. 657-664.

[4] C. Candan, “An Efficient Filtering Structure for Lagrange Interpolation”, Signal Processing Letters, IEEE, vol. 14, 2007, pp. 17-19.

[5] P. Duda, M. Jaworski, L. Pietruczuk, R., Scherer, M. Korytkowski and M. Gabryel, “On the Application of Fourier Series Density Estimation for Image Classification Based on Feature Description”, Proceedings of KICSS’2013, 7-9 November, Cracow, Poland, 2013, pp. 81-89.

[6] M. Fleck, “Some Defects in Finite-Difference Edge Finders”, Trans. on Pattern Analysis and Machine Intelligence, IEEE, vol. 14, no. 3, 1992, pp. 337-345.

[7] M. Gabryel, R. K. Nowicki, M. Woźniak, and W. M. Kempa, “Genetic cost optimization of the GI/M/1/N finite-buffer queue with a single vacation policy”, ICAISC'2013, Lecture Notes in Artificial Intelligence, Springer Verlag, PART II, vol. 7895, 2013, pp. 12-23.

[8] M. Gabryel, M. Woźniak and R. K. Nowicki, “Creating learning sets for control systems using an evolutionary method”, ICAISC'2012, Lecture Notes in Computer Science, Springer Verlag, vol. 7269, 2012, pp. 206-213.

[9] G. K. Knopf, and A. Sangole, “Interpolating scattered data using 2D self-organizing feature maps”, Graphical Models, no. 66, 2004, pp. 50-69.

[10] P. Najgebauer, M. Gabryel, M. Korytkowski and R. Scherer, “Tree Representation of Image Key Point Descriptors”, Proceedings of KICSS’2013, 7-9 November, Cracow, Poland, 2013, pp. 402-412.

[11] A. Nowak A. and M. Woźniak M, “Algorithm for optimization of the active module by the use of genetic algorithm”, Acta Mechanica Slovaca, Košice, Slovakia, no. 3C, 2008, pp. 307-316.

[12] T. Poggio and T. Vetter, “Recognition and Structure from one 2D Model View: Observations on Prototypes, Object Classes and Symmetries”, Technical Report, A.I.Memo, no. 1347, C.B.I.P., 1992, no. 69, pp. 1-25.

[13] S. P. Raya and J. K. Udupa, “Shape-Based Interpolation of Multidimensional Object”, Trans. on Pattern Analysis and Machine Intelligence, IEEE, 1990, vol. 9, no. 1, pp. 32-42.

[14] J. R. Rossignac and J.J.Kim, “Computing and visualizing pose-interpolating 3D motions”, Computer-Aided Design, no. 33, 2001, pp. 279-291.

[15] H. E. Salzer, “Lagranian interpolation at Chebyshev points $x_{n,v}$=cos(vπ/n), v= 0(1)n; some unnoted advantages”, The Computer Journal, vol. 15, no. 2, 1997, pp. 156-159.

[16] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data”, Proceedings of the ACM national conference, New York, USA, 1968, pp. 517-524.

[17] F. Stein, and G. Medioni, “Structural Indexing: Efficient 2-D Object Recognition”, Trans. on Pattern Analysis and Machine Intelligence, IEEE, vol. 14, no. 12, 1992, pp. 1198-1204.

[18] J. V. Stone, “Object Recognition using Spatiotemporal Signatures”, Vision Research, vol. 38, no. 7, 1998, pp. 947-951.

[19] Q. M. Tieng and W. W. Boles, “Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation”, Trans. on Pattern Analysis and Machine Intelligence, IEEE, vol. 19, no. 8, 1997, pp. 910-916.

[20] T. Vetter and T. Poggio, “Linear Object Classes and Image Synthesis From a Single Example Image”, Trans. on Pattern Analysis and Machine Intelligence, IEEE, vol. 19, no. 7, 1997, pp. 910-916.

[21] Y. Xu, “Lagrange Interpolation on Chebyshev Points of Two Variables”, Journal of Approximation Theory, no. 87, 1996, pp. 220-238.

[22] H. S. Yang and S. U. Lee, “Recognition of 2D Object Contours Using Starting-Point-Independent Wavelet Coefficient Matching”, Visual Communication and Image Representational, vol. 9, no. 2, 1998, pp. 171-181.

[23] M. Woźniak, W. M. Kempa, M. Gabryel, R. K. Nowicki and Z. Shao, "On applying evolutionary computation methods to optimization of vacation cycle costs in finite-buffer queue", ICAISC’2014, Lecture Notes in Artificial Intelligence, Springer-Verlag, PART I, vol. 8467, 2014, pp. 480-491.