Open Access Journal of Computer Science and Engineering
Volume 1, Issue 1, 2024, PP: 22-32
www.aytinpublications.com

Research Article

Extrapolation and Interpolation of Data in Numerical Analysis

Dariusz Jacek Jakóbczak*

Department of Electronics and Computer Science, Koszalin University of Technology, Sniadeckich 2, 75-453 Koszalin, Poland

***Corresponding Author:** Dariusz Jacek Jakóbczak, Department of Electronics and Computer Science, Koszalin University of Technology, Sniadeckich 2, 75-453 Koszalin, Poland.

Abstract

Proposed method is dealing with multi-dimensional data modeling, extrapolation and interpolation using the set of high-dimensional feature vectors. Identification of handwriting, signature, faces or fingerprints need data modeling and each model of the pattern is built by a choice of characteristic key points and multi-dimensional modeling functions. Novel modeling via nodes combination and parameter γ as *N*-dimensional function enables data parameterization and interpolation for feature vectors. Multi-dimensional data is modeled and interpolated via different functions for each feature: polynomial, sine, cosine, tangent, cotangent, logarithm, exponent, arc sin, arc cos, arc tan, arc cot or power function.

Keywords: Image Retrieval, Pattern Recognition, Data Modeling, Vector Interpolation, Feature Reconstruction, Curve Modeling

Aricle Information

Recieved: 13 September 2024

Accepted: 28 September 2024

Published: 30 September 2024

Cite this article as:

Dariusz Jacek Jakóbczak, Extrapolation and Interpolation of Data in Numerical Analysis. Open Access Journal of Computer Science and Engineering, 2024;1(1); 22-32.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Introduction

The idea of paper is connected with different curve modeling for the same set of curve points (nodes). The problem of multidimensional data modeling appears in many branches of science and industry. Image retrieval, data reconstruction, object identification or pattern recognition are still the open problems in artificial intelligence and computer vision. The paper is dealing with these questions via modeling of high-dimensional data for applications of image segmentation in image retrieval and recognition tasks. Handwriting based author recognition offers a huge number of significant implementations which make it an important research area in pattern recognition. There are so many possibilities and applications of the recognition algorithms that implemented methods have to be concerned on a single problem: retrieval, identification, verification or recognition. This paper is concerned with two parts: image retrieval and recognition tasks. Image retrieval is based on modeling of unknown features via combination of *N*-dimensional functions for each feature. In the case of biometric writer recognition, each person is represented by the set of modeled letters or symbols. The sketch of proposed method consists of three steps: first handwritten letter or symbol must be modeled by a vector of features (*N*-dimensional data), then compared with unknown letter and finally there is a decision of identification. Author recognition of handwriting and signature is based on the choice of feature vectors and modeling functions. So high-dimensional data interpolation in handwriting identification [20] is not only a pure mathematical problem but important task in pattern recognition and artificial intelligence such as: biometric recognition, personalized handwriting recognition [3-5], automatic forensic document examination [6,7], classification of ancient manuscripts [8]. Also writer recognition [9] in monolingual handwritten texts is an extensive area of study and the methods independent from the language are well-seen [10-13]. Proposed method represents language-independent and text-independent approach because it identifies the author via a set of letters or symbols from the sample.

Writer recognition methods in the recent years are going to various directions [14-18]: writer recognition using multi-script handwritten texts, introduction of new features, combining different types of features, studying the sensitivity

of character size on writer identification, investigating writer identification in multi-script environments, impact of ruling lines on writer identification, model perturbed handwriting, methods based on run-length features, the edge-direction and edge-hinge features, a combination of codebook and visual features extracted from chain code and polygonized representation of contours, the autoregressive coefficients, codebook and efficient code extraction methods, texture analysis with Gabor filters and extracting features, using Hidden Markov Model [19] or Gaussian Mixture Model [1]. So hybrid soft computing is essential: no method is dealing with writer identification via *N*-dimensional data modeling or interpolation and multidimensional points comparing as it is presented in this paper. The paper wants to approach a problem of curve interpolation and shape modeling by characteristic points in handwriting identification [2]. Proposed method relies on nodes combination and functional modeling of curve points situated between the basic set of key points. The functions that are used in calculations represent whole family of elementary functions with inverse functions: polynomials, trigonometric, cyclometric, logarithmic, exponential and power function. Nowadays methods apply mainly polynomial functions, for example Bernstein polynomials in Bezier curves, splines [25] and NURBS. But Bezier curves don't represent the interpolation method and cannot be used for example in signature and handwriting modeling with characteristic points (nodes). Numerical methods [21-23] for data interpolation are based on polynomial or trigonometric functions, for example Lagrange, Newton, Aitken and Hermite methods. These methods have some weak sides and are not sufficient for curve interpolation in the situations when the curve cannot be build by polynomials or trigonometric functions [24].

This paper presents novel method of high-dimensional interpolation in hybrid soft computing and takes up method of multidimensional data modeling. The method requires information about data (image, object, curve) as the set of *N*-dimensional feature vectors. So this paper wants to answer the question: how to retrieve the image using *N*-dimensional feature vectors and to recognize a handwritten letter or symbol by a set of high-dimensional nodes via hybrid soft computing?

Multidimensional Modeling of Feature Vectors

Proposed method is computing (interpolating) unknown (unclear, noised or destroyed) values of features between two successive nodes (*N*-dimensional vectors of features) using hybridization of mathematical analysis and numerical methods, Calculated values (unknown or noised features such as coordinates, colors, textures or any coefficients of pixels, voxels and doxels or image parameters) are interpolated and parameterized for real number $\alpha_i \in [0;1]$ ($i = 1,2,\ldots N\text{-}1$) between two successive values of feature. This method uses the combinations of nodes (*N*-dimensional feature vectors) $p_1=(x_1,y_1,\ldots,z_1)$, $p_2=(x_2,y_2,\ldots,z_2),\ldots,\ p_n=(x_n,y_n,\ldots z_n)$ as $h(p_1,p_2,\ldots,p_m)$ and $m=1,2,\ldots n$ to interpolate unknown value of feature (for example *y*) for the rest of coordinates:

$c_1 = \alpha_1 x_k + (1-\alpha_1)\cdot x_{k+1},\ldots\ldots\ c_{N-1} = \alpha_{N-1}\cdot z_k + (1-\alpha_{N-1})\cdot z_{k+1}$, $k = 1,2,\ldots n\text{-}1$,

$c = (c_1,\ldots, c_{N-1}),\quad \alpha = (\alpha_1,\ldots, \alpha_{N-1}),\quad \gamma_i = F_i(\alpha_i) \in [0;1]$, $i = 1,2,\ldots N\text{-}1$

$$y(c) = \mathrm{g}\cdot y_k + (1-\mathrm{g})y_{k+1} + \mathrm{g}(1-\mathrm{g})\cdot h(p_1,p_2,\ldots,p_m) \qquad (1)$$

$\alpha_i \in [0;1],\ \gamma = F(\alpha) = F(\alpha_1,\ldots,\alpha_{N-1}) \in [0;1]$.

Then N-1 features $c_1,\ldots, c_{N-1}$ are parameterized by $\alpha_1,\ldots, \alpha_{N-1}$ between two nodes and the last feature (for example *y*) is interpolated via formula (1). Of course there can be calculated $x(c)$ or $z(c)$ using (1). Two examples of *h* (when $N = 2$) computed for MHR method [26] with good features because of orthogonal rows and columns at Hurwitz-Radon family of matrices:

$$h(p_1,p_2) = \frac{y_1}{x_1}x_2 + \frac{y_2}{x_2}x_1 \qquad (2)$$

or

$$h(p_1,p_2,p_3,p_4) = \frac{1}{x_1^2+x_3^2}(x_1x_2y_1 + x_2x_3y_3 + x_3x_4y_1 - x_1x_4y_3) + \frac{1}{x_2^2+x_4^2}(x_1x_2y_2 + x_1x_4y_4 + x_3x_4y_2 - x_2x_3y_4)$$

The simplest nodes combination is

$$h(p_1,p_2,\ldots,p_m) = 0 \qquad (3)$$

and then there is a formula of interpolation:

$$y(c) = \mathrm{g}\cdot y_i + (1-\mathrm{g})y_{i+1}.$$

Formula (1) gives the infinite number of calculations for unknown feature determined by choice of *F* and *h*. Nodes combination is the individual feature of each modeled data. Coefficient $\gamma=F(\alpha)$ and nodes combination *h* are key factors in data interpolation and object modeling.

N-Dimensional Functions in Modeling

Unknown values of features, settled between the nodes, are computed using (1). Key question is dealing with coefficient γ. The simplest way of calculation means $h = 0$ and $\gamma_i = \alpha_i$. Then proposed method represents a linear interpolation. Each interpolation requires specific values of α_i and γ in (1) depends on parameters $\alpha_i \in [0;1]$:

$\gamma = F(\alpha),\ F:[0;1]^{N-1}\rightarrow[0;1],\ F(0,\ldots,0) = 0,\ F(1,\ldots,1) = 1$

and *F* is strictly monotonic for each α_i separately. Coefficient γ_i are calculated using appropriate function and choice

of function is connected with initial requirements and data specifications. Different values of coefficients γ_i are connected with applied functions $F_i(\alpha_i)$. These functions $\gamma_i = F_i(\alpha_i)$ represent the examples of modeling functions for $\alpha_i \in [0;1]$ and real number $s > 0$, $i = 1,2,\ldots N\text{-}1$. Each function is applied for different modelling:

$\gamma_i=\alpha_i^s$, $\gamma_i=sin(\alpha_i^s\cdot\pi/2)$, $\gamma_i=sin^s(\alpha_i\cdot\pi/2)$, $\gamma_i=1\text{-}cos(\alpha_i^s\cdot\pi/2)$, $\gamma_i=1\text{-}cos^s(\alpha_i\cdot\pi/2)$, $\gamma_i=tan(\alpha_i^s\cdot\pi/4)$, $\gamma_i=tan^s(\alpha_i\cdot\pi/4)$, $\gamma_i=log_2(\alpha_i^s+1)$, $\gamma_i=log_2^s(\alpha_i+1)$, $\gamma_i=(2^\alpha-1)^s$, $\gamma_i=2/\pi\cdot arcsin(\alpha_i^s)$, $\gamma_i=(2/\pi\cdot arcsin\alpha_i)^s$, $\gamma_i=1\text{-}2/\pi\cdot arccos(\alpha_i^s)$, $\gamma_i=1\text{-}(2/\pi\cdot arccos\alpha_i)^s$, $\gamma_i=4/\pi\cdot arctan(\alpha_i^s)$, $\gamma_i=(4/\pi\cdot arctan\alpha_i)^s$, $\gamma_i=ctg(\pi/2-\alpha_i^s\cdot\pi/4)$, $\gamma_i=ctg^s(\pi/2\text{-}\alpha_i\cdot\pi/4)$, $\gamma_i=2\text{-}4/\pi\cdot arcctg(\alpha_i^s)$, $\gamma_i=(2\text{-}4/\pi\cdot arcctg\alpha_i)^s$ or any strictly monotonic function between points (0;0) and (1;1). For example interpolations of function $y=2^x$ for $N = 2$, $h = 0$ and $\gamma = \alpha^s$ with $s = 0.8$ (Fig.1) is much better than linear interpolation.

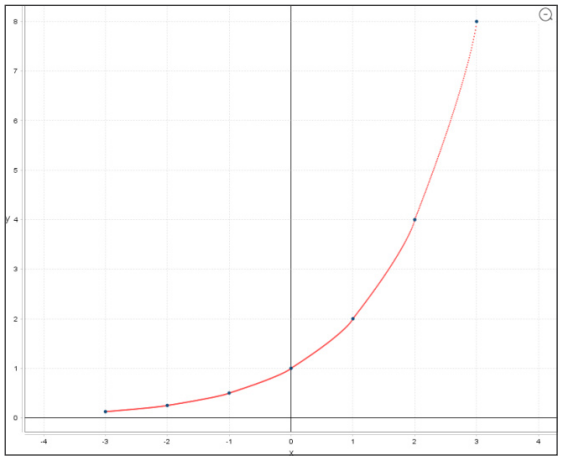

Figure 1. *Two-dimensional modeling of function $y=2^x$ with seven nodes and $h=0$, $\gamma=\alpha^{0.8}$.*

Functions γ_i are strictly monotonic for each variable $\alpha_i \in [0;1]$ as $\gamma = F(\alpha)$ is N-dimensional modeling function, for example:

$$\gamma = \frac{1}{N-1}\sum_{i-1}^{N-1}\gamma_i, \quad \gamma = \prod_{i=1}^{N-1}\gamma_i$$

and every monotonic combination of γ_i such as

$$\gamma = F(\alpha),\ F:[0;1]^{N-1}\rightarrow[0;1],\ F(0,\ldots,0) = 0,\ F(1,\ldots,1) = 1.$$

For example when $N = 3$ there is a bilinear interpolation:

$$\gamma_1 = \alpha_1,\ \gamma_2 = \alpha_2,\ \gamma = \tfrac{1}{2}(\alpha_1 + \alpha_2) \qquad \textbf{(4)}$$

or a bi-quadratic interpolation:

$$\gamma_1 = \alpha_1^2,\ \gamma_2 = \alpha_2^2,\ \gamma = \tfrac{1}{2}(\alpha_1^2 + \alpha_2^2) \qquad \textbf{(5)}$$

or a bi-cubic interpolation:

$$\gamma_1 = \alpha_1^3,\ \gamma_2 = \alpha_2^3,\ \gamma = \tfrac{1}{2}(\alpha_1^3 + \alpha_2^3) \qquad \textbf{(6)}$$

or others modeling functions γ. Choice of functions γ_i and value s depends on the specifications of feature vectors and individual requirements. What is very important: two data sets (for example a handwritten letter or signature) may have the same set of nodes (feature vectors: pixel coordinates, pressure, speed, angles) but different h or γ results in different interpolations (Fig.2-4). Here are three examples of reconstruction (Fig.2-4) for $N = 2$ and four nodes: (-1.5;-1), (1.25;3.15), (4.4;6.8) and (8;7). Formula of the curve is not given. Algorithm of proposed retrieval, interpolation and modeling consists of five steps: first choice of nodes p_i (feature vectors), then choice of nodes combination $h(p_1,p_2,\ldots,p_m)$, choice of modeling function $\gamma = F(\alpha)$, determining values of $\alpha_i \in [0;1]$ and finally the computations (1).

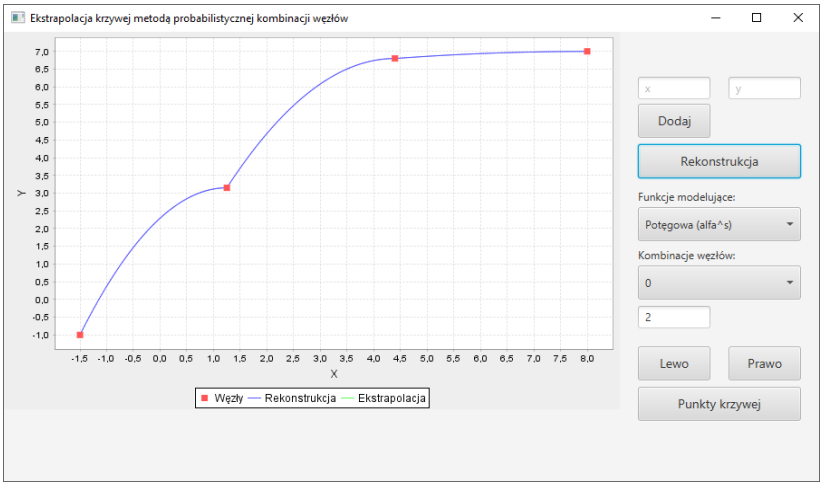

Figure 2. *2D modeling for $\gamma = \alpha^2$ and $h = 0$.*

And other interpolations for the same set of nodes:

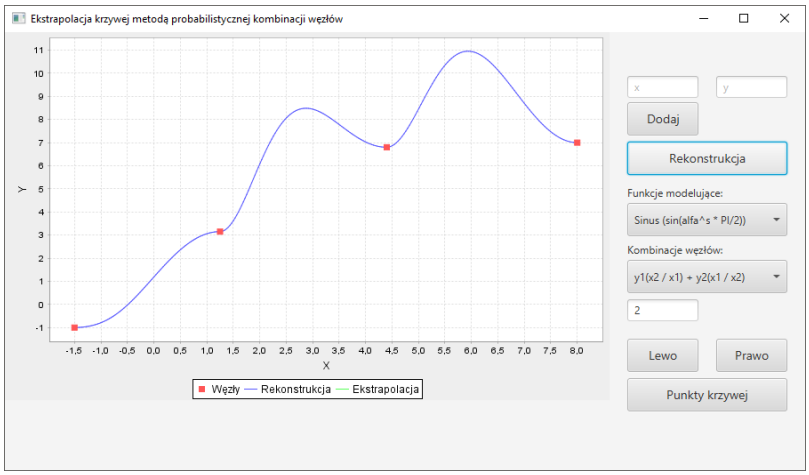

Figure 3. *2D reconstruction for γ = sin(α²·π/2) and h in (2).*

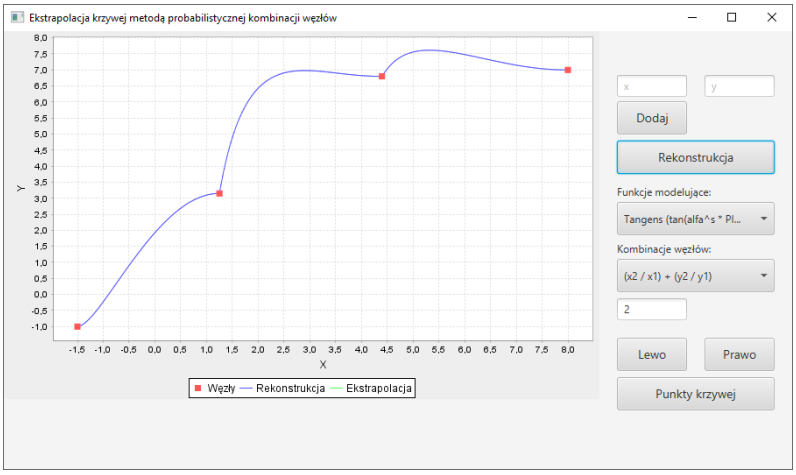

Figure 4. *2D interpolation for γ = tan(α²·π/4) and h = (x₂/x₁)+ (y₂/y₁).*

So there are different data reconstructions with different modeling functions. As it can be observed, there is one extremum between two nodes for modeling with h ≠ 0 (Fig.3-4). Comparing with polynomial or spline interpolations, there is one very important question: **how to avoid extremum between each pair of nodes and how to minimize interpolation error?** Generally current methods do not answer this key question. Nowadays methods of interpolations rely mainly on polynomials, especially on cubic splines. It means that there are interpolation polynomials *W(x)* of degree 3 for every range of two successive interpolation nodes (x_i,y_i) and (x_{i+1},y_{i+1}). This method of cubic splines is C^2 class – this fact is very important in many applications of cubic interpolation. But second important feature of this method is interpolation error for function *f(x)*:

$$|f(x)-W(x)| \le 5M|(x-x_i)(x-x_{i+1})|,$$
$$M = \sup_{x\in[a,b]}|f''(x)|.$$

So interpolation error depends on second derivative in the range of nodes [*a;b*] and this value cannot be estimated in general. Cubic spline can have extremum and may differ from interpolated function *f(x)* very much. Also interpolation polynomial $W_n(x)$ of degree *n* (Lagrange or Newton) for *n*+1 nodes $(x_0,y_0), (x_1,y_1) \dots (x_n,y_n)$ is connected with unpredictable error in general with calculations of derivative rank *n*+1:

$$|f(x)-W_n(x)| \le \frac{M_{n+1}}{(n+1)!}|(x-x_o)(x-x_1)...(x-x_n)|,$$
$$M_{n+1} = \sup_{x\in[a,b]}|f^{(n+1)}(x)|.$$

Proposed method with $h = 0$ and $\alpha \in [0;1]$ represents formulas as convex combinations of nodes' coordinates:

$$x(\alpha) = \alpha\cdot x_k + (1-\alpha)x_{k+1}, \qquad y(\alpha) = \gamma_k \cdot y_k + (1-\gamma_k)y_{k+1}.$$

and interpolation error in general between two nodes looks as follows:

$$\varepsilon_k \le |y_{k+1} - y_k|.$$

Proposed method is dealing with such significant features:

- no extremum between two nodes;
- interpolation error does not depend on the value of derivative in the nodes or outside the nodes (even if derivative does not exist);

- interpolated function can be smooth in the nodes (class C^1);
- reconstruction of the function that much differs from the shape of polynomial, and not only function but any curve, also closed;
- extrapolation is calculated with the same formulas for $\alpha \notin [0;1]$;
- the idea of linear interpolation is applied for other modeling functions, not only $\gamma = \alpha$[1];
- convexity between the nodes is fixed using two modeling functions:

  $\gamma_k = \alpha^s$ or $\gamma_k = sin(\alpha^s \cdot \pi/2)$ with real parameter $s > 0$.

These two kinds of modeling functions are the simplest function, chosen via many calculations as follows:

- $\gamma_k = \alpha^s$ if convexity is not changing between the nodes (x_k,y_k) and (x_{k+1},y_{k+1});
- $\gamma_k = sin(\alpha^s \cdot \pi/2)$ if convexity is changing between the nodes (x_k,y_k) and (x_{k+1},y_{k+1}).

Theorem If

1. There are given nodes of continuous function $y = f(x)$: (x_0,y_0), (x_1,y_1) … (x_n,y_n), $n \geq 2$;
2. There are formulas to calculate values between the nodes:

$$x(\alpha) = \alpha \cdot x_k + (1-\alpha)x_{k+1}, \quad y(\alpha) = \gamma_k \cdot y_k + (1-\gamma_k)y_{k+1}.$$

   $\alpha \in [0;1]$, $k = 2,3\ldots n\text{-}1$, $\gamma_k = \alpha^s$ or $\gamma_k = sin(\alpha^s \cdot \pi/2)$ with real parameter $s > 0$;
3. Three successive nodes are monotonic, for example let's assume:

$y_0 > y_1 > y_2$ or $y_0 < y_1 < y_2$.

Then there is the method of 2D curve interpolation and extrapolation such as:

T.1: There is no extremum between two successive nodes – interpolated function is monotonic in the range of two nodes.

T.2: Interpolated curve is class C^0 (continuous) or C^1 (continuous and smooth).

T.3: Interpolation error does not depend on the value of derivative in the nodes or outside the nodes (even if derivative does not exist).

T.4: Convexity between two nodes (x_k,y_k) and (x_{k+1},y_{k+1}) is fixed using modeling functions $\gamma_k = \alpha^s$ (if convexity is not changing) or $\gamma_k = sin(\alpha^s \cdot \pi/2)$ (if convexity is changing).

T.5: Extrapolation is calculated with the same formulas for $\alpha \notin [0;1]$.

Proof

T.1: Convex combination to calculate $x(\alpha)$ and $y(\alpha)$ between two nodes with strictly monotonic function γ_k gives us monotonic interpolation of the curve with no extremum between two nodes.

T.2: Interpolated curve is class C^0 (continuous) just from definition of $x(\alpha)$ and $y(\alpha)$. Also smooth interpolation between nodes is achieved with the same. Only smooth function in the inner nodes must be proved. Here is shown how to achieve smooth function in the inner nodes – let's assume then $y_k \neq y_{k+1}$ for each k. If $y_k = y_{k+1}$ for any k, then according to T.1 there must be the simplest linear interpolation between nodes (x_k,y_k) and (x_{k+1},y_{k+1}) and interpolated curve is not smooth in nodes (x_k,y_k) and (x_{k+1},y_{k+1}).

For first three monotonic nodes (x_0,y_0), (x_1,y_1) and (x_2,y_2) there are calculations to fix parameter s for modeling function γ_1 between nodes (x_0,y_0) and (x_2,y_2) interpolating node (x_1,y_1) inside:

$$\alpha = \frac{x_2 - x_1}{x_2 - x_0} \in (0;1), \qquad t = \frac{y_2 - y_1}{y_2 - y_0} \in (0;1).$$

If convexity is not changing between (x_0,y_0) and (x_2,y_2), then $\gamma_1 = \alpha^s$ and $s = \log_\alpha t$.

If convexity is changing between (x_0,y_0) and (x_2,y_2), then $\gamma_1 = sin(\alpha^s \cdot \pi/2)$ and

$$s = \log_\alpha \left(\frac{2}{\pi}\arcsin t\right).$$

A1 (beginning of the loop in algorithm for $k = 2,3\ldots n\text{-}1$): Having modeling function γ_1 between nodes (x_0,y_0) and (x_2,y_2), it is possible for any $\alpha^* \to 0$ calculate

$$x(\alpha^*) = \alpha^* \cdot x_0 + (1-\alpha^*)x_2, \qquad y(\alpha^*) = \gamma_1 \cdot y_0 + (1-\gamma_1)y_2.$$

Then left difference quotient c is computed in the node (x_2,y_2):

$$c = \frac{y_2 - y(\alpha^*)}{x_2 - x(\alpha^*)}.$$

Of course if value of derivative in (x_2,y_2) is known, $c = f'(x_2) \neq 0$. Then parameter u is fixed to obtain left (c) and right difference quotient equal in (x_2,y_2) - it means smooth in this node. If y_3 preserves the same monotonicity like y_2 and y_1 ($y_1 > y_2 > y_3$ or $y_1 < y_2 < y_3$) then

$$u = 1 - c(1-\alpha^*)\frac{x_3 - x_2}{y_3 - y_2}.$$

If y_3 does not preserve the same monotonicity like y_2 and y_1 then (because of different sign of left and right difference quotient)

$$u = 1 + c(1-\alpha^*)\frac{x_3 - x_2}{y_3 - y_2}.$$

And as it was: if convexity is not changing between (x_2,y_2) and (x_3,y_3), then $\gamma_2 = \alpha^s$ and

$$s = \log_{\alpha^*} u.$$

If convexity is changing between (x_2,y_2) and (x_3,y_3), then $\gamma_2 = sin(\alpha^s \cdot \pi/2)$ and

$$s = \log_{\alpha^*}(\frac{2}{\pi}\arcsin u).$$

So smooth interpolation function in the node (x_2,y_2) is achieved. And smooth interpolation for next range of nodes (x_3,y_3) and (x_4,y_4) is starting like loop **A1** for k=3. And so on till last range of nodes (x_{n-1},y_{n-1}) and (x_n,y_n) for $k = n$-1 in **A1**.

T.3: According to T.1 – interpolation error between two nodes for each k is equal:

$$\varepsilon_k \le |y_{k+1} - y_k|.$$

T.4: These modeling functions are the simplest functions to achieve convexity changing or not.

T.5: Extrapolation left of first node (x_0,y_0) is done with modeling function γ_1 and α>1. Extrapolation right of last node (x_n,y_n) is done with modeling function γ_{n-1} and α<0. Then modeling function γ_{n-1} must have domain with α<0. If not, there is possibility to define:

$$x(\alpha) = \alpha \cdot x_{k+1} + (1-\alpha)x_k, \quad y(\alpha) = \gamma_k \cdot y_{k+1} + (1-\gamma_k)y_k$$

This theorem describes main features of proposed method.

Image Retrieval via High-dimensional Feature Reconstruction

After the process of image segmentation and during the next steps of retrieval, recognition or identification, there is a huge number of features included in N-dimensional feature vector. These vectors can be treated as “points” in N-dimensional feature space. For example in artificial intelligence there is a high-dimensional search space (the set of states that can be reached in a search problem) or hypothesis space (the set of hypothesis that can be generated by a machine learning algorithm). This paper is dealing with multidimensional feature spaces that are used in computer vision, image processing and machine learning.

Having monochromatic (binary) image which consists of some objects, there is only 2-dimensional feature space (x_i,y_i) – coordinates of black pixels or coordinates of white pixels. No other parameters are needed. Thus any object can be described by a contour (closed binary curve). Binary images are attractive in processing (fast and easy) but don’t include important information. If the image has grey shades, there is 3-dimensional feature space (x_i,y_i,z_i) with grey shade z_i. For example most of medical images are written in grey shades to get quite fast processing. But when there are color images (three parameters for RGB or other color systems) with textures or medical data or some parameters, then it is N-dimensional feature space. Dealing with the problem of classification learning for high-dimensional feature spaces in artificial intelligence and machine learning (for example text classification and recognition), there are some methods: decision trees, k-nearest neighbors, perceptrons, naïve Bayes or neural networks methods. All of these methods are struggling with the curse of dimensionality: the problem of having too many features. And there are many approaches to get less number of features and to reduce the dimension of feature space for faster and less expensive calculations. This paper aims at inverse problem to the curse of dimensionality: dimension N of feature space (i.e. number of features) is unchanged, but number of feature vectors (i.e. “points” in N-dimensional feature space) is reduced into the set of nodes. *So the main problem is as follows: how to fix the set of feature vectors for the image and how to retrieve the features between the “nodes”?* This paper aims in giving the answer of this question.

Grey Scale Image Retrieval Using 3d Method

Binary images are just the case of 2D points (x,y): 0 or 1, black or white, so retrieval of monochromatic images is done for the closed curves (first and last node are the same) as the contours of the objects for N = 2 and examples as Fig.1-4. Grey scale images are the case of 3D points (x,y,s) with s as the shade of grey. So the grey scale between the nodes $p_1=(x_1,y_1,s_1)$ and $p_2=(x_2,y_2,s_2)$ is computed with $\gamma = F(\alpha) = F(\alpha_1,\alpha_2)$ as (1) and for example (4)-(6) or others modeling functions γ_i. As the simple example two successive nodes of the image are: left upper corner with coordinates $p_1=(x_1,y_1,2)$ and right down corner $p_2=(x_2,y_2,10)$. The image retrieval with the grey scale 2-10 between p_1 and p_2 looks as follows for a bilinear interpolation (4):

2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10

Figure 5. *Reconstructed grey scale numbered at each pixel.*

The feature vector of dimension N = 3 is called a voxel.

3.2 Color image retrieval

Color images in for example RGB color system (r,g,b) are the set of points (x,y,r,g,b) in a feature space of dimension N = 5. There can be more features, for example texture t, and then one pixel (x,y,r,g,b,t) exists in a feature space of dimension N = 6. But there are the sub-spaces of a feature space of dimension $N_1 < N$, for example (x,y,r),

(*x,y,g*), (*x,y,b*) or (*x,y,t*) are points in a feature sub-space of dimension $N_1 = 3$. Reconstruction and interpolation of color coordinates or texture parameters is done like in section 3.1 for dimension $N = 3$. Appropriate combination of α_1 and α_2 leads to modeling of color *r*,*g*,*b* or texture *t* or another feature between the nodes. And for example (*x,y,r,t*), (*x,y,g,t*), (*x,y,b,t*)) are points in a feature sub-space of dimension N_1=4 called doxels. Appropriate combination of α_1, α_2 and α_3 leads to modeling of texture *t* or another feature between the nodes. For example color image, given as the set of doxels (*x,y,r,t*), is described for coordinates (*x,y*) via pairs (*r,t*) interpolated between nodes (x_1,y_1,2,1) and (x_2,y_2,10,9) as follows:

2,1	3,1	4,1	5,1	6,1	7,1	8,1	9,1	10,1
2,2	3,2	4,2	5,2	6,2	7,2	8,2	9,2	10,2
2,3	3,3	4,3	5,3	6,3	7,3	8,3	9,3	10,3
2,4	3,4	4,4	5,4	6,4	7,4	8,4	9,4	10,4
2,5	3,5	4,5	5,5	6,5	7,5	8,5	9,5	10,5
2,6	3,6	4,6	5,6	6,6	7,6	8,6	9,6	10,6
2,7	3,7	4,7	5,7	6,7	7,7	8,7	9,7	10,7
2,8	3,8	4,8	5,8	6,8	7,8	8,8	9,8	10,8
2,9	3,9	4,9	5,9	6,9	7,9	8,9	9,9	10,9

Figure 6. *Color image with color and texture parameters (r,t) interpolated at each pixel.*

So dealing with feature space of dimension *N* and using novel method there is no problem called "the curse of dimensionality" and no problem called "feature selection" because each feature is important. There is no need to reduce the dimension *N* and no need to establish which feature is "more important" or "less important". Every feature that depends from N_1-1 other features can be interpolated (reconstructed) in the feature sub-space of dimension $N_1 < N$ via proposed method. But having a feature space of dimension *N* and using author's method there is another problem: how to reduce the number of feature vectors and how to interpolate (retrieve) the features between the known vectors (called nodes). Difference between two given approaches (the curse of dimensionality with feature selection and author's interpolation) can be illustrated as follows. There is a feature matrix of dimension *N* x *M*: *N* means the number of features (dimension of feature space) and *M* is the number of feature vectors (interpolation nodes) – columns are feature vectors of dimension *N*. One approach (Fig.7): the curse of dimensionality with feature selection wants to eliminate some rows from the feature matrix and to reduce dimension *N* to $N_1 < N$. Second approach (Fig.8) for this method wants to eliminate some columns from the feature matrix and to reduce dimension *M* to $M_1 < M$.

2	2	2	2	2	2	2	2	2		2	2	2	2	2	2	2	2	2
2	3	3	3	3	3	3	3	3		2	3	3	3	3	3	3	3	3
2	3	4	4	4	4	4	4	4		2	3	4	4	4	4	4	4	4
2	3	4	5	5	5	5	5	5		2	3	4	5	5	5	5	5	5
2	3	4	5	6	6	6	6	6	→	2	3	4	5	6	6	6	6	6
2	3	4	5	6	7	7	7	7		2	3	4	5	6	7	7	7	7
2	3	4	5	6	7	8	8	8										
2	3	4	5	6	7	8	9	9										
2	3	4	5	6	7	8	9	10										

Figure 7. *The curse of dimensionality with feature selection wants to eliminate some rows from the feature matrix and to reduce dimension N.*

2	2	2	2	2	2	2	2	2		2	2	2	2	2	2	2
2	3	3	3	3	3	3	3	3		2	3	3	3	3	3	3
2	3	4	4	4	4	4	4	4		2	3	4	4	4	4	4
2	3	4	5	5	5	5	5	5		2	3	4	5	5	5	5
2	3	4	5	6	6	6	6	6	→	2	3	4	5	6	6	6
2	3	4	5	6	7	7	7	7		2	3	4	5	6	7	7
2	3	4	5	6	7	8	8	8		2	3	4	5	6	7	8
2	3	4	5	6	7	8	9	9		2	3	4	5	6	7	8
2	3	4	5	6	7	8	9	10		2	3	4	5	6	7	8

Figure 8. *Proposed method wants to eliminate some columns from the feature matrix and to reduce dimension M.*

So after feature selection (Fig.7) there are nine feature vectors (columns): $M = 9$ in a feature sub-space of dimension $N_1 = 6 < N$ (three features are fixed as less important and reduced). But feature elimination is a very unclear matter. And what to do if every feature is denoted as meaningful and then no feature is to be reduced? For this method (Fig.8) there are seven feature vectors (columns): $M_1 = 7 < M$ in a feature space of dimension $N = 9$. Then

no feature is eliminated and the main problem is dealing with interpolation or extrapolation of feature values, like for example image retrieval (Fig.5-6).

Recognition Tasks via High-Dimensional Feature Vectors' Interpolation

The process of biometric recognition and identification consists of three parts: pre-processing, image segmentation with feature extraction and recognition or verification. Pre-processing is a common stage for all methods with binarization, thinning, size standardization. Proposed online approach is based on 2D curve modeling and multi-dimensional feature vectors' interpolation. Feature extraction gives the key points (nodes as *N*-dimensional feature vectors) that are used in curve reconstruction and identification. Proposed method enables signature and handwriting recognition, which is used for biometric purposes, because human signature or handwriting consists of non-typical curves and irregular shapes (for example Fig.2-4). The language does not matter because each symbol is treated as a curve. This process of recognition consists of three parts:

1. Before recognition – continual and never-ending building the data basis: patterns' modeling – choice of nodes combination, function (1) and values of features (pen pressure, speed, pen angle etc.) appearing in high dimensional feature vectors for known signature or handwritten letters of some persons in the basis;
2. Feature extraction: unknown author – fixing the values in feature vectors for unknown signature or handwritten words: *N*-dimensional feature vectors (*x,y,p,s,a,t*) with *x,y*-points' coordinates, *p*-pen pressure, *s*-speed of writing, *a*- pen angle or any other features *t*;
3. The result: recognition or identification - comparing the results of interpolation for known patterns from the data basis with features of unknown object.

Signature Modeling and Multidimensional Recognition

Human signature or handwriting consists mainly of non-typical curves and irregular shapes. So how to model two-dimensional handwritten characters via author's method? Each model has to be described (1) by the set of nodes, nodes combination *h* and a function $\gamma=F(\alpha)$ for each letter. Other features in multi-dimensional feature space are not visible but used in recognition process (for example *p*-pen pressure, *s*-speed of writing, *a*-pen angle). Less complicated models can take $h(p_1,p_2,\ldots,p_m) = 0$ and then the formula of interpolation (1) looks as follows:

$$y(c) = \gamma \cdot y_i + (1-\gamma) y_{i+1}. \qquad (7)$$

Formula (7) represents the simplest linear interpolation if $\gamma = \alpha$. Here are some examples of non-typical curves and irregular shapes as the whole signature or a part of signature, reconstructed via proposed method for $y=2^x$ and seven nodes (x,y) like Fig.1:

Lp.	x	y
1	-3.0	0.125
2	-2.0	0.25
3	-1.0	0.5
4	0.0	1.0
5	1.0	2.0
6	2.0	4.0
7	3.0	8.0

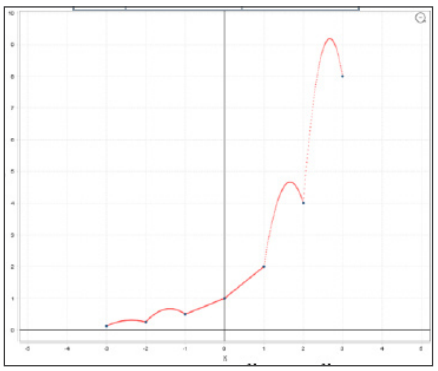

Figure 9. *2D interpolation for* $\gamma = \alpha^s,\ s = 1,\ h = \frac{y_1}{x_1}x_2 + \frac{y_2}{x_2}x_1$

And two other interpolations for the same set of nodes:

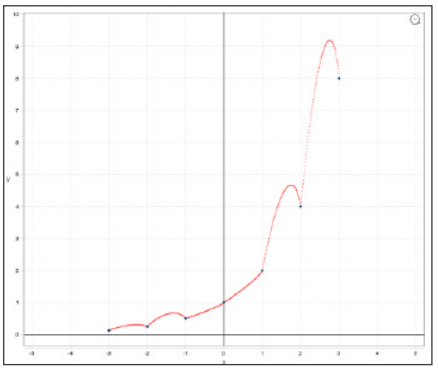

Figure 10. *2D modeling for* $\gamma = \alpha^s,\ s = 0{,}8,\ h = \frac{y_1}{x_1}x_2 + \frac{y_2}{x_2}x_1.$

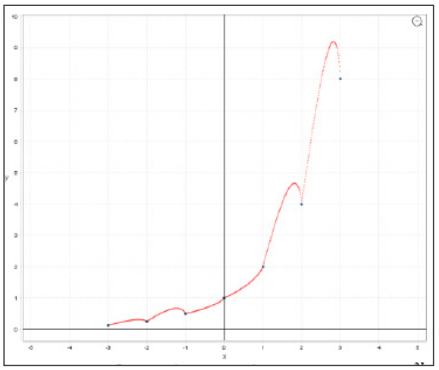

Figure 11. *2D reconstruction for* $\gamma = log_2(\alpha^s + 1),\ s = 0{,}8,\ h = \frac{y_1}{x_1}x_2 + \frac{y_2}{x_2}x_1.$

So there are different data reconstructions with different modeling functions. Other interpolations for the same set of nodes and combination $h=0$ are as follows:

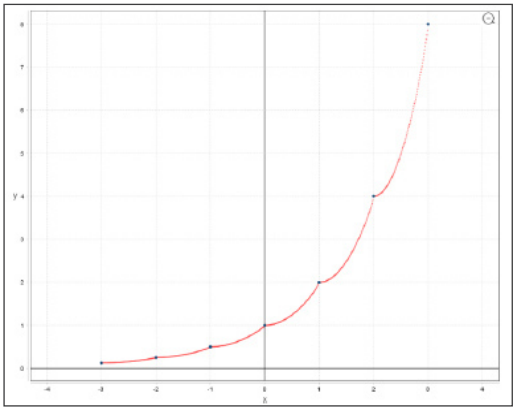

Fig. 12. 2D modeling for $\gamma = sin^{s}(\alpha \cdot \frac{\pi}{2}),\ s = 0{,}8,\ h = 0.$

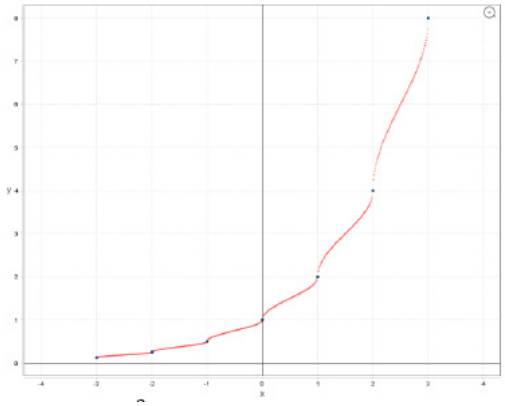

Figure 13. *2D modeling for* $\gamma = 1 - \frac{2}{\pi}\arccos(\alpha^{s}),\ s = 0{,}5,\ h = 0.$

Fig.9-13 are two-dimensional subspace of N-dimensional feature space, for example (x,y,p,s,a,t) when $N = 6$. If the recognition process is working "offline" and features p-pen pressure, s-speed of writing, a- pen angle or another feature t are not given, the only information before recognition is situated in x,y-points' coordinates.

After pre-processing (binarization, thinning, size standarization), feature extraction is second part of biometric identification. Choice of characteristic points (nodes) for unknown letter or handwritten symbol is a crucial factor in object recognition. The range of coefficients x has to be the same like the x range in the basis of patterns. When the nodes are fixed, each coordinate of every chosen point on the curve (x_0^c,y_0^c), $(x_1^c,y_1^c),\ldots,(x_M^c,y_M^c)$ is accessible to be used for comparing with the models. Then modeling function $\gamma = F(\alpha)$ and nodes combination h have to be taken from the basis of modeled letters to calculate appropriate second coordinates $y_i^{(j)}$ of the pattern $\boldsymbol{S}_j$ for first coordinates x_i^c, $i = 0,1,\ldots,M$. After interpolation it is possible to compare given handwritten symbol with a letter in the basis of patterns. Comparing the results of this interpolation for required second coordinates of a model in the basis of patterns with points on the curve (x_0^c,y_0^c), $(x_1^c,y_1^c),\ldots,(x_M^c,y_M^c)$, one can say if the letter or symbol is written by person P1, P2 or another. The comparison and decision of recognition is done via minimal distance criterion. Curve points of unknown handwritten symbol are: (x_0^c,y_0^c), $(x_1^c,y_1^c),\ldots,(x_M^c,y_M^c)$. The criterion of recognition for models $\boldsymbol{S}_j = \{(x_0^c,y_0^{(j)}),\ (x_1^c,y_1^{(j)}),\ldots,\ (x_M^c,y_M^{(j)}),\ j=0,1,2,3\ldots K\}$ is given as:

$$\sum_{i=0}^{M}\left|y_i^c - y_i^{(j)}\right| \to \min \quad \text{or} \quad \sqrt{\sum_{i=0}^{M}\left|y_i^c - y_i^{(j)}\right|^2} \to \min . \qquad \textbf{(8)}$$

Minimal distance criterion helps us to fix a candidate for unknown writer as a person from the model $\boldsymbol{S}_j$ in the basis. If the recognition process is "online" and features p-pen pressure, s-speed of writing, a- pen angle or some feature t are given, then there is more information in the process of author recognition, identification or verification in a feature space (x,y,p,s,a,t) of dimension $N = 6$ or others. Some person may know how the signature of another man looks like (for example Fig.2-4 or Fig.9-13), but other extremely important features p,s,a,t are not visible. Dimension N of a feature space may be very high, but this is no problem. As it is illustrated (Fig.7-8) the problem connected with the curse of dimensionality with feature selection does not matter. There is no need to fix which feature is less important and can be eliminated. Every feature is very important and each of them can be interpolated between the nodes using author's high-dimensional interpolation. For example pressure of the pen p differs during the signature writing and p is changing for particular letters or fragments of the signature. Then feature vector (x,y,p) of dimension $N_1 = 3$ is dealing with p interpolation at the point (x,y) via modeling functions (4)-(6) or others. If angle of the pen a differs during the signature writing and a is changing for particular letters or fragments of the signature, then feature vector (x,y,a) of dimension $N_1 = 3$ is dealing with a interpolation at the point (x,y) via modeling functions (4)-(6) or others. If speed of the writing s differs during the signature writing and s is changing for particular letters or fragments of the signature, then feature vector (x,y,s) of dimension $N_1 = 3$ is dealing with s interpolation at the point (x,y) via modeling functions (4)-(6) or others. This 3D interpolation is the same like in section 3.1 grey scale image retrieval but for selected pairs (α_1,α_2) – only for the points of signature between $(x_1,y_1,2)$ and $(x_2,y_2,10)$:

2	3	0	0	0	0	0	0	0
0	0	4	5	0	0	0	0	0
0	0	0	0	6	0	0	0	0
0	0	0	0	0	7	0	0	0
0	0	0	0	0	0	8	0	0
0	0	0	0	0	0	0	9	0
0	0	0	0	0	0	0	0	10
0	0	0	0	0	0	0	0	10
0	0	0	0	0	0	0	0	10

Figure 14. *Reconstructed speed of the writing s at the pixels of signature.*

If a feature sub-space is dimension N_1 = 4 and feature vector is for example (*x,y,p,s*), then 4D interpolation is the same like in section 3.2 color image retrieval but for selected pairs (α_1,α_2) – only for the points of signature between (x_1,y_1,2,1) and (x_2,y_2,10,9):

2,1	3,1	0	0	0	0	0	0	0
0	0	4,2	5,2	0	0	0	0	0
0	0	0	0	6,3	0	0	0	0
0	0	0	0	0	7,4	0	0	0
0	0	0	0	0	0	8,5	0	0
0	0	0	0	0	0	0	9,6	0
0	0	0	0	0	0	0	0	10,7
0	0	0	0	0	0	0	0	10,8
0	0	0	0	0	0	0	0	10,9

Figure 15. Reconstructed pen pressure *p* and speed of the writing *s* as (*p*,*s*) at the pixels of signature.

If a feature sub-space is dimension N_1 = 5 and feature vector is for example (*x,y,p,s,a*), then 5D interpolation is the same like in section 3.2 color image retrieval but for selected pairs (α_1,α_2) – only for the points of signature between (x_1,y_1,2,1,30) and (x_2,y_2,10,9,60):

2,1,30	3,1,30	0	0	0	0	0	0	0
0	0	4,2,32	5,2,34	0	0	0	0	0
0	0	0	0	6,3,37	0	0	0	0
0	0	0	0	0	7,4,43	0	0	0
0	0	0	0	0	0	8,5,45	0	0
0	0	0	0	0	0	0	9,6,46	0
0	0	0	0	0	0	0	0	10,7,53
0	0	0	0	0	0	0	0	10,8,56
0	0	0	0	0	0	0	0	10,9,60

Figure 16. *Reconstructed pen pressure p, speed of the writing s and angle a as (p,s,a) at the pixels of signature.*

Fig.14-16 are the examples of denotation for the features that are not visible during the signing or handwriting but very important in the process of "online" recognition, identification or verification. Even if from technical reason or other reasons only some points of signature or handwriting (feature nodes) are given in the process of "online" recognition, identification or verification, the values of features between nodes are computed via multidimensional author's interpolation like for example between (x_1,y_1,2) and (x_2,y_2,10) on Fig.14, between (x_1,y_1,2,1) and (x_2,y_2,10,9) on Fig.15 or between (x_1,y_1,2,1,30) and (x_2,y_2,10,9,60) on Fig.16. Reconstructed features are compared with the features in the basis of patterns like parameter *y* in (8) and appropriate criterion gives the result.

So persons with the parameters of their signatures are allocated in the basis of patterns. The curve does not have to be smooth at the nodes because handwritten symbols are not smooth. The range of coefficients *x* has to be the same for all models because of comparing appropriate coordinates *y*. Every letter or a part of signature is modeled via three factors: the set of high-dimensional feature nodes, modeling function $\gamma=F(\alpha)$ and nodes combination *h*. These three factors are chosen individually for each letter or a part of signature therefore this information about modeled curves seems to be enough for specific multidimensional curve interpolation and handwriting identification. What is very important, novel *N*-dimensional modeling is independent of the language or a kind of symbol (letters, numbers, characters or others). One person may have several patterns for one handwritten letter or signature. Summarize: every person has the basis of patterns for each handwritten letter or symbol, described by the set of feature nodes, modeling function $\gamma = F(\alpha)$ and nodes combination *h*. Whole basis of patterns consists of models $\boldsymbol{S}_j$ for j = 0,1,2,3…*K*. Proposed interpolation is used for parameterization and reconstruction of curves in the plane.

Conclusion

The autor's method enables interpolation and modeling of high-dimensional data using features' combinations and different coefficients γ: polynomial, sinusoidal, cosinusoidal, tangent, cotangent, logarithmic, exponential, arc sin, arc cos, arc tan, arc cot or power function. Functions for γ calculations are chosen individually at each data modeling and it is treated as *N*-dimensional function: γ depends on initial requirements and features' specifications. Novel method leads to data interpolation as handwriting or signature identification and image retrieval via discrete set of feature vectors in *N*-dimensional feature space. So this method makes possible the combination of two important problems: interpolation and modeling in a matter of image retrieval or writer identification. Main features of the method are: this interpolation develops a linear interpolation in multidimensional feature spaces into other functions as *N*-dimensional functions; nodes combination and coefficients γ are crucial in the process of data parameterization and interpolation: they are computed individually for a single feature; modeling of closed curves.

References

1. Schlapbach, A., Bunke, H.: Off-line writer identification using Gaussian mixture models. In: International Conference on Pattern Recognition, pp. 992–995 (2006)

2. Bulacu, M., Schomaker, L.:Text-independent writer identification and verification using textural and allographic features. IEEE Trans. Pattern Anal. Mach. Intell. 29 (4), 701–717 (2007)

3. Djeddi, C., Souici-Meslati, L.: A texture based approach for Arabic writer identification and verification. In: International Conference on Machine and Web Intelligence, pp. 115–120 (2010)

4. Djeddi, C., Souici-Meslati, L.: Artificial immune recognition system for Arabic writer identification. In: International Symposium on Innovation in Information and Communication Technology, pp. 159–165 (2011)

5. Nosary, A., Heutte, L., Paquet, T.: Unsupervised writer adaption applied to handwritten text recognition. Pattern Recogn. Lett. 37 (2), 385–388 (2004)

6. Van, E.M., Vuurpijl, L., Franke, K., Schomaker, L.: The WANDA measurement tool for forensic document examination. J. Forensic Doc. Exam. 16, 103–118 (2005)

7. Schomaker, L., Franke, K., Bulacu, M.: Using codebooks of fragmented connected- component contours in forensic and historic writer identification. Pattern Recogn. Lett. 28 (6), 719–727 (2007)

8. Siddiqi, I., Cloppet, F., Vincent, N.: Contour based features for the classification of ancient manuscripts. In: Conference of the International Graphonomics Society, pp. 226–229 (2009)

9. Garain, U., Paquet, T.: Off-line multi-script writer identification using AR coefficients. In: International Conference on Document Analysis and Recognition, pp. 991–995 (2009)

10. Bulacu, M., Schomaker, L., Brink, A.: Text-independent writer identification and verification on off-line Arabic handwriting. In: International Conference on Document Analysis and Recognition, pp. 769–773 (2007)

11. Ozaki, M., Adachi, Y., Ishii, N.: Examination of effects of character size on accuracy of writer recognition by new local arc method. In: International Conference on Knowledge-Based Intelligent Information and Engineering Systems, pp.1170–1175 (2006)

12. Chen, J., Lopresti, D., Kavallieratou, E.: The impact of ruling lines on writer identification. In: International Conference on Frontiers in Handwriting Recognition, pp. 439–444 (2010)

13. Chen, J., Cheng, W., Lopresti, D.: Using perturbed handwriting to support writer identification in the presence of severe data constraints. In: Document Recognition and Retrieval, pp. 1–10 (2011)

14. Galloway, M.M.: Texture analysis using gray level run lengths. Comput. Graphics Image Process. 4 (2), 172–179 (1975)

15. Siddiqi, I., Vincent, N.: Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. Pattern Recogn. Lett. 43 (11), 3853–3865 (2010)

16. Ghiasi, G., Safabakhsh, R.: Offline text-independent writer identification using codebook and efficient code extraction methods. Image and Vision Computing 31, 379–391 (2013)

17. Shahabinejad, F., Rahmati, M.: A new method for writer identification and verification based on Farsi/Arabic handwritten texts, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), pp. 829–833 (2007)

18. Schlapbach, A., Bunke, H.: A writer identification and verification system using HMM based recognizers, Pattern Anal. Appl. 10, 33–43 (2007)

19. Schlapbach, A., Bunke, H.: Using HMM based recognizers for writer identification and verification, 9th Int. Workshop on Frontiers in Handwriting Recognition, pp. 167–172 (2004)

20. Marti, U.-V., Bunke, H.: The IAM-database: an English sentence database for offline handwriting recognition, Int. J. Doc. Anal. Recognit. 5, 39–46 (2002)

21. Collins II, G.W.: Fundamental Numerical Methods and Data Analysis. Case Western Reserve University (2003)

22. Chapra, S.C.: Applied Numerical Methods. McGraw-Hill (2012)

23. Ralston, A., Rabinowitz, P.: A First Course in Numerical Analysis – Second Edition. Dover Publications, New York (2001)

24. Zhang, D., Lu, G.: Review of Shape Representation and Description Techniques. Pattern Recognition 1(37), 1-19 (2004)

25. Schumaker, L.L.: Spline Functions: Basic Theory. Cambridge Mathematical Library (2007)

26. Jakóbczak, D.J.: 2D Curve Modeling via the Method of Probabilistic Nodes Combination - Shape Representation, Object Modeling and Curve Interpolation-Extrapolation with the Applications. LAP Lambert Academic Publishing, Saarbrucken (2014)