Clareus Scientific Science and Engineering
Volume 2 Issue 2 February 2025
DOI: 10.70012/CSSE.02.013
ISSN: 3065-1182

Compression and Decompression of Digital-Ink Handwriting Using Sparse Gaussian Process Regression and Dynamic Programming

Citation: Hiroyuki Fujioka., et al. "Compression and Decompression of Digital-Ink Handwriting Using Sparse Gaussian Process Regression and Dynamic Programming". Clareus Scientific Science and Engineering 2.2 (2025): 03-11.

Article Type: Research Article

Received: January 29, 2025

Published: February 14, 2025

Jinya Yano[1] and Hiroyuki Fujioka[2]*

[1]Graduate School of Engineering, Fukuoka Institute of Technology, Japan

[2]Department of Information Management, Fukuoka Institute of Technology, Japan

***Corresponding Author:** Hiroyuki Fujioka, Department of Information Management, Fukuoka Institute of Technology, 3-30-1 Wajiro-Higashi, Higashi-ku, Fukuoka 811-0295, Japan.

Abstract

This study addresses the challenges of compressing and decompressing digital-ink data captured from handwritten inputs on a device- such as pen-tablet, etc. We propose a novel method leveraging sparse Gaussian process regression (GPR) to compress digital-ink data into a compact kernel matrix using pseudo-inputs. To enhance compression accuracy and reconstruction fidelity, we integrate a dynamic programming (DP) strategy for optimal pseudo-input selection. Experimental results validate the proposed method by demonstrating its effectiveness and significant compression ratios.

Keywords: digital-ink; handwriting; compression; decompression; sparse Gaussian process regression; pseudo-inputs; optimal selection; dynamic programming

Abbreviations

GPR: Gaussian process regression.
DP: Dynamic Programming.

Introduction

Despite the increasing digitalization of communication, handwriting remains integral to personal expression and professional interactions. Modern devices such as tablets, smartphones, and electronic whiteboards facilitate handwriting-based input, offering users a natural and intuitive interface [1]. Such an input, stored as 'digital-ink', consists of data points sampled from traced curves at fixed intervals. However, as handwriting sessions lengthen or sampling rates increase, the resulting data volume poses challenges for storage, transmission, and processing.

To address these challenges, several methods have been studied to compress and decompress digital-ink data, such as orthogonal polynomial approximations [2] and piecewise-linear approximations [3]. Mieno et al. [4] introduced a B-spline-based method to represent digital-ink using control points. Despite their utility, these methods struggle with complex handwriting styles, often requiring a high number of coefficients or control points, thereby increasing computational costs. Sparse modelling techniques, such as Lasso [5, 6], offer an alternative approach, but their reliance on predefined dictionaries limits their adaptability to diverse handwriting patterns [7].

Recent advancements in Gaussian Process Regression (GPR) provide a promising solution to these limitations (see e.g. [8]). As a non-parametric probabilistic model, GPR excels at capturing the underlying structure of high-dimensional and nonlinear data. In particular, sparse GPR, introduced by Snelson and Ghahramani [9], extends this framework by approximating the full covariance matrix with a small set of pseudo-inputs, significantly reducing computational complexity. This method ensures efficient compression and accurate reconstruction, making it ideal for large handwriting datasets.

In this research, we present a new approach to addressing the challenges of digital-ink data compression and reconstruction using sparse GPR. Our method introduces an innovative framework that condenses extensive digital-ink information into a compact set of pseudo-inputs, effectively tackling the inherent issues highlighted in previous studies [2-4]. By leveraging the power of sparse GPR, we ensure that the compressed representation retains the essential features of the original digital-ink while significantly reducing data size. To further refine the accuracy of compression and decompression, we incorporate a dynamic programming (DP) methodology for the optimal selection of dominant pseudo-inputs. The DP approach prioritizes the most critical data points, enhancing both compression efficiency and reconstruction fidelity. The DP algorithm constructs a weighted directed acyclic graph (DAG) from the digital-ink dataset and identifies the minimal error path to select the key representative points. This hybrid integration of sparse GPR and DP sets our method apart, offering a robust and efficient solution for digital-ink data handling while maintaining a high degree of fidelity to the original manuscript.

The remainder of this paper is structured as follows: We first introduce the theoretical foundations of GPR and basic idea of digital-ink compression and decompression using GPR. Next we elaborate on the sparse GPR methodology, and then detail the DP-based pseudo-input selection algorithm. In addition, we present experimental results demonstrating the efficacy of the proposed methods, and finally concludes the paper with insights into future research directions.

Preliminaries: Basic Idea on Digital-Ink Compression and Decompression via Gaussian Processing Regression

As preliminaries, we present the Gaussian Process Regression (GPR) [8] and some basic idea on digital-ink compression and decompression using GPR.

Now, we suppose that a set of digital-ink data is measured by pen-tablet device, etc. and are stored as

$$D = \{(t, y_i) : t_i \in [x_{ini}, x_{fin}], y_i \in \boldsymbol{R}^2, i = 1,2,\cdots, N\}, \tag{1}$$

where the data is selected at equal intervals, t_{i+1}-t_i = *const*. Also, $y_i = [X_i, Y_i]^T$ corresponds to the two-dimensional spatial coordinates of the ink stroke. Note here that the writing process is considered as a single-stroke activity with uniform thickness. Additionally, the mean of y is normalized to be 0. We then consider to represent a set of digital-ink data D by employing an idea of GPR. Namely, our task is to construct a model to estimate a function $f(t)$ with an input $t \in [t_{ini}, t_{fin}]$ (i.e. time) and digital-ink $y = [X, Y]^T \in \boldsymbol{R}^2$. For the sake of simplicity, we here assume that each element in $y(t) \in \boldsymbol{R}^2$, i.e. $X(t)$ and $Y(t)$, is treated independently, and using notational abuse, $y(t)$ is treated as a scalar with the understanding that they represent one of the two elements.

Here, we suppose that the function $f(t)$ is generated from a Gaussian process (GP), i.e.

$$f(t) \sim GP(0, k(t, t')) \tag{2}$$

Let $\boldsymbol{y}$ be a vector defined by

$$\boldsymbol{y} = [y_1, y_2, \cdots, y_N]^{\mathrm{T}}. \tag{3}$$

Then, we have

$$y \sim N(\mathbf{0}, \boldsymbol{K}), \tag{4}$$

where $K \in R^{N \times N}$ denotes a kernel matrix given as

$$\boldsymbol{K} = \begin{bmatrix} k(t_1,t_1) & k(t_1,t_2) & \cdots & k(t_1,t_N) \\ k(t_2,t_1) & k(t_2,t_2) & \cdots & k(t_2,t_N) \\ \vdots & & \ddots & \vdots \\ k(t_N,t_1) & k(t_N,t_2) & \cdots & k(t_N,t_N) \end{bmatrix}. \tag{5}$$

Here we use a Gaussian kernel as the kernel function $k(t_i, t_j)$, i.e.

$$k(t_i, t_j) = \exp\left(-\frac{|t_i - t_j|^2}{\theta_1}\right) + \theta_2 \delta(t_i, t_j), \tag{6}$$

where θ_i for i =1,2 denote hyperparameters and $\delta(\cdot,\cdot)$ the Kronecker delta.

Once we obtain the kernel matrix $\boldsymbol{K}$ in (5) (or $\boldsymbol{t} = [t_1, t_2, \cdots, t_N]^T \in \boldsymbol{R}^N$ for computing $\boldsymbol{K}$) as compressed data of digital-ink, we can readily decompress to the digital-ink by computing as expectation of posterior as follows.

Now letting (t^*, y^*) for $t^* \in [t_{ini}, t_{fin}]$ be a digital-ink data decompressed from the kernel matrix $\boldsymbol{K}$ in (5), the joint distribution of $\boldsymbol{y}$ in (3) and y^*according to the prior follows a Gaussian distribution as

$$\begin{pmatrix} \boldsymbol{y} \\ y^* \end{pmatrix} \sim N\left(\mathbf{0}, \begin{pmatrix} \boldsymbol{K} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^T & k_{**} \end{pmatrix}\right) \tag{7}$$

with

$$k^* = [k(t^*, t_1), \cdots, k(t^*, t_N)]^T, \tag{8}$$

$$k^{**} = k(t^*, t^*). \tag{9}$$

Let $p(y^* | D, t^*)$ be a posterior distribution of y^*corresponding to some time $t^* \in [t_{ini}, t_{fin}]$. Then, it can be shown that $p(y^* | D, t^*)$ is written as

$$p(y^*|D, t^*) = N(\boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{y}, k_{**} - \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{k}_*). \tag{10}$$

For decompressing from $\boldsymbol{K}$ in (5) to the digital-ink, we thus have only to iteratively compute an expectation of posterior $p(y^* | D, t^*)$ in (10), i.e.

$$E[y^*|D, t^*] = \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{y}. \tag{11}$$

This compression method is straightforward as it only requires storing the kernel matrix $\boldsymbol{K}$ in (5) or $\boldsymbol{t} = [t_1, t_2, \cdots, t_N]^T$ for computing $\boldsymbol{K}$. However, in this method, the decompression typically demands $\boldsymbol{O}(N^3)$ time due to the inversion of $\boldsymbol{K}$, making it impractical when N becomes extremely large (e.g., when the time interval of handwriting data is extensive). Then, one of natural approaches to address this issue could be the use of Cholesky decomposition [10]. But, this method also requires all elements of the kernel matrix $\boldsymbol{K}$, rendering it unsuitable for large datasets as the computational complexity grows exponentially with the number of data N.

Sparse GPR-based Compression and Decompression

As a method to address the issues on standard GPR discussed in the above, one promising approach may be to adopt the concept of sparse GPR [9]. By selecting a small number of the so-called "inducing points", sparse GPR approximates the full kernel matrix of the Gaussian process, significantly reducing computational complexity. Here, we present a method for compressing and decompressing digital-ink data using sparse GPR.

Now, let $\boldsymbol{z} \in \boldsymbol{R}^M$ be a vector consisting of $M(\ll N)$ pseudo-input points $z_i \in [t_{ini}, t_{fin}]$, known as inducing points, defined as

$$z = [z_1, z_2, \cdots, z_M]^T. \tag{12}$$

While the specifics of the derivation are beyond the scope of this paper (see [9] for details), it can be shown that the kernel matrix in (5) is approximated as

$$\boldsymbol{K} \approx \boldsymbol{K}_{NM}\boldsymbol{K}_{MM}^{-1}\boldsymbol{K}_{NM}^T + \boldsymbol{\Lambda}. \tag{13}$$

Here, $\boldsymbol{K}_{NM} \in \boldsymbol{R}^{N \times M}$ denotes a matrix with elements $k(t_i, z_j)$ for $i = 1, \cdots, N$ and $j = 1, \cdots, M$. Also, $\boldsymbol{K}_{MM} \in \boldsymbol{R}^{M \times M}$ denotes a matrix with $k(z_i, z_j)$ for $i, j = 1, \cdots, M$ as an element. $\boldsymbol{\Lambda}$ is a diagonal matrix defined by

$$\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \cdots, \lambda_N\} \tag{14}$$

with

$$\lambda_n = k(t_n, t_n) - \boldsymbol{k}_{nM}\boldsymbol{K}_{MM}^{-1}\boldsymbol{k}_{nM}^T, \qquad n = 1, \cdots, N. \tag{15}$$

Here, $k_{nM} \in R^{1 \times M}$ denotes n-th row vector of $\boldsymbol{K}_{NM}$. In (13), the number of operations required for calculating the kernel matrix is proportional to the number of pseudo-inputs M. In other words, in the sparse GPR, the smaller the number of pseudo-inputs, the fewer the number of operations required for calculating the kernel matrix, resulting in sparsity.

On the other hand, let $\boldsymbol{u} \in \boldsymbol{R}^M$ be a vector defined by

$$\boldsymbol{u} = [u_1, u_2, \cdots, u_M]^T, \tag{16}$$

where u_i denotes output values corresponding to each pseudo-input point z_i, i.e. $u_i = f(z_i)$. Similar to (4), we have

$$\boldsymbol{u} \sim N(\boldsymbol{0}_M, \boldsymbol{K}_{MM}). \tag{17}$$

Using Bayes' rule, the posterior distribution over $\boldsymbol{u}$, i.e. $p(\boldsymbol{u}|\boldsymbol{y})$, is given by

$$p(\boldsymbol{u}|\boldsymbol{y}) = N\left(\hat{\boldsymbol{u}}, \hat{\Sigma}_{\boldsymbol{u}}\right). \tag{18}$$

Here, $\hat{\boldsymbol{u}}$ and $\hat{\Sigma}_{\boldsymbol{u}}$ are given as

$$\hat{\boldsymbol{u}} = \boldsymbol{K}_{MM}\boldsymbol{Q}_{MM}^{-1}\boldsymbol{K}_{NM}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{y}, \tag{19}$$

$$\hat{\Sigma}_{\boldsymbol{u}} = \boldsymbol{K}_{MM}^{-1}\boldsymbol{Q}_{MM}\boldsymbol{K}_{MM}^{-1} \tag{20}$$

with

$$\boldsymbol{Q}_{MM} = \boldsymbol{K}_{MM} + \boldsymbol{K}_{NM}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{NM}. \tag{21}$$

Now, let (t^*, y^*) for $t^* \in [t_{ini}, t_{fin}]$ be a new data. Then, the digital-ink decompression can be achieved by using a posterior distribution $p(y^* \mid D, t^*)$ of y^* corresponding to some time $t^* \in [t_{ini}, t_{fin}]$, just as in the same idea of standard GPR. Using (18), it can be shown that $p(y^* \mid D, t^*)$ is expressed as

$$p(y^*|t^*, D) = N\left(\hat{y}_*, \hat{\sigma}_{y_*}^2\right) \tag{22}$$

with

$$\hat{y}_* = \boldsymbol{k}_*^T\boldsymbol{K}_{MM}^{-1}\hat{\boldsymbol{u}}, \tag{23}$$

$$\hat{\sigma}_{y_*}^2 = k_{**} - \boldsymbol{k}_*^T(\boldsymbol{K}_{MM}^{-1} - \boldsymbol{Q}_{MM}^{-1})\boldsymbol{k}_*. \tag{24}$$

In order to decompress from $\boldsymbol{K}_{MM}$ and $\boldsymbol{K}_{NM}$ in (13), we have only to iteratively compute an expectation of posterior $p(y^* \mid t^*, D)$ in (22), i.e.

$$E[y^*|D,t^*] = \boldsymbol{k}_*^T \boldsymbol{K}_{MM}^{-1} \hat{\boldsymbol{u}} \tag{25}$$

for t^*sampled between t_{ini} and t_{fin}.

By comparing these with the basic ideas on digital-ink compression and decompression described above, we see that as long as we compress the digital-ink as $\boldsymbol{K}_{MM} \in \boldsymbol{R}^{M \times M}$, we can decompress it by decompress it by (25). In addition, the computation of kernel matrix $\boldsymbol{K}_{MM}$ results in a smaller $M \times M$ kernel matrix compared to standard GPR. This reduces the computational complexity for computing the inverse matrix, i.e. $\boldsymbol{K}_{MM}^{-1}$ in (25) to $\boldsymbol{O}(M^3)$. Thus, this proposed method using Sparse GPR approach may become a widely used implementation for digital-ink with large-scale datasets, due to its efficiency in handling such datasets.

Optimal Pseudo-Input Point Selection via Dynamic Programming approach

To enhance the accuracy of Sparse GPR-based compression and decompression method, we introduce a dynamic programming (DP) approach inspired by [3] for the optimal selection of dominant pseudo-input points, i.e. $\boldsymbol{z}$ in (12). This approach prioritizes essential data points, improving both compression efficiency and data integrity.

Let G be a weighted directed acyclic graph (DAG) [11], constructed from the digital-ink dataset D in (1). In addition, letting $V(G)$ and $E(G)$ be a set of vertices and edges of graph G, we have

$$\begin{cases} V(G) = \{v_i | 1 \le i \le N\} \\ E(G) = \{(v_i, v_j) | 1 \le i < j \le N\} \end{cases} \tag{26}$$

Here, each vertex v_i, $1 \le i \le N$ corresponds to the digital-ink position data $y_i = [X_i\ Y_i]^T$, $1 \le i \le N$, respectively.

Also, let $\gamma(v_i, v_j)(\ge 0)$ be the weight between vertices v_i and v_j, defined as

$$\gamma(v_i, v_j) = \sum_{k=i}^{j-1} \|(X_{k+1}, Y_{k+1}) - (X_k, Y_k)\|_2 - \|(X_j, Y_j) - (X_i, Y_i)\|_2 , \tag{27}$$

where $\|\cdot\|_2$ denotes the L_2 norm.

The DAG, G, has distinct starting and ending points, which align with y_1 and y_N. The aim is to identify a path in G that starts at y_1, ends at y_N, and consists of M vertices representing dominant data point. This path should minimize the total weight defined by (27). The existence and determination of such a path via DP are discussed in [3].

Now, let $D_{M,j}$ be the minimum total weight of a path from v_1 to v_j that include M vertices. Then, we initially set $D_{M,j}$ for $M = 2$ (i.e., $D_{2,j}$) as

$$D_{2,j} = \begin{cases} \infty & \text{if } j = 1 \\ \gamma(v_1, v_j) & \text{if } 1 < j \le N \end{cases} . \tag{28}$$

For $M \ge 3$, $D_{M,j}$ is iteratively computed as

$$D_{M,j} = \begin{cases} min_{M-1 \le i < j}\{D_{M-1,i} + \gamma(v_i, v_j)\} & \text{if } j \ge M \\ \infty & \text{if otherwise} \end{cases} . \tag{29}$$

To optimally select M dominant pseudo-input points, we thus have only to compute $D_{M,N}$, which represents the minimum cumulative error. The resulting path provides the desired pseudo-input points, minimizing the error function and ensuring optimal compression.

Experimental Studies

We here evaluate the performance of the proposed sparce GPR-based digital-ink compression and decompression methods through experimental studies.

Experimental System

An overview of the experimental system is shown in Figure 1. The system comprises a PC running Vine Linux 3.0 and a Wacom PL-550-10D0 pen tablet. This setup captures and stores handwriting input as digital-ink data D, which represents 2D positions on the XY-plane sampled at 20 [ms] intervals. The proposed method based on sparce GPR-based compression and decompression is implemented by employing MATLAB on a PC running Windows 10 Education. The hardware specifications of the PC include an Intel Core i9-7900X CPU and 128 GB of RAM.

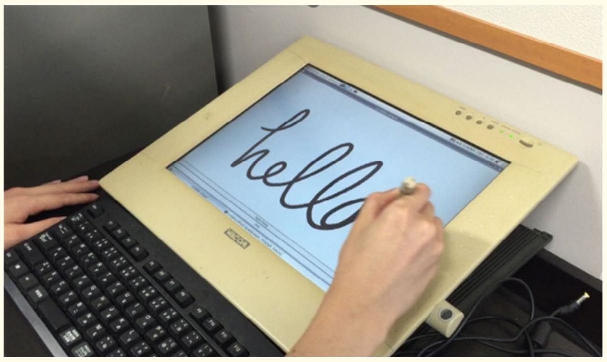


Figure 1: An overview of the experimental system.

Experimental Results

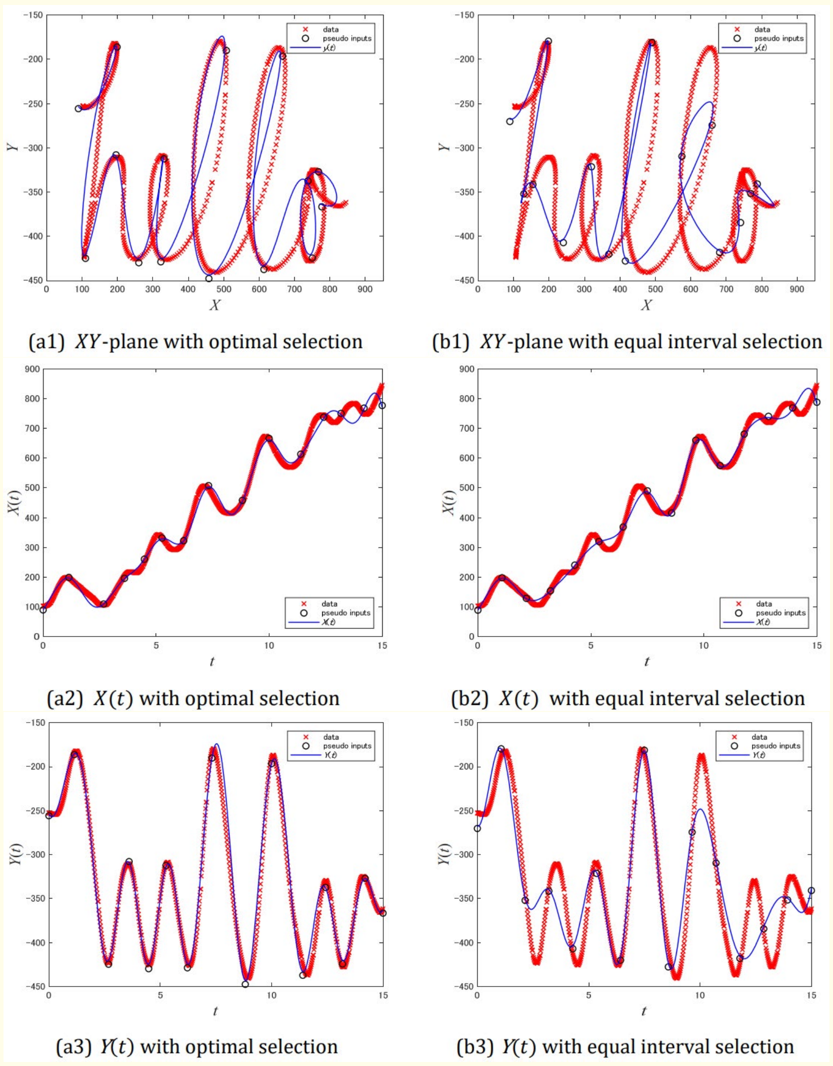
As an example, we present the results for the word “hello” written in a single stroke. The digital-ink D in (1) is measured and stored using the experimental system in Figure 1, where the total number of data points is $N = 751$. Additionally, the time range is set as $t_{ini} = 0$ [s] and $t_{fin} = 15$ [s].

Figures 2 and 3 illustrate the compressed and decompressed results, where we set M as $M = 15$ and $M = 20$ in Figure 2 and Figure 3, respectively. Also, the panels in these figures are organized as follows:

- Panels (a1), (a2), and (a3) in Figures 2 and 3:
  These panels show the decompressed digital-ink using the pseudo-inputs selected by the dynamic programming (DP) approach:
  - ➢ Figures 2 and 3 (a1) illustrates the decompressed 2D path of digital-ink of $y(t) = [X(t)\ Y(t)]^T$ (black lines), overlaid with the measured digital-ink data (red cross marks) and pseudo-inputs (blue circles).
  - ➢ (a2) and (a3) of Figures 2 and 3 show $X(t)$ and $Y(t)$ of the digital-ink $y(t)$, respectively.
- Panels (b1), (b2), and (b3) in Figures 2 and 3:
  These panels show the results when using an equivalent number of pseudo-inputs ($M = 15$ or $M = 20$) but arranged uniformly at regular intervals.
  - ➢ Figures 2 and 3 (b1) illustrates the decompressed 2D path (XY plane).
  - ➢ (b2) and (b3) of Figures 2 and 3 show $X(t)$ and $Y(t)$ of the digital-ink $y(t)$, respectively.

For both configurations, the hyperparameters θ_1 and θ_2 in (6) are set as $\theta_1 = 0.1$ and $\theta_2 = 0.01$.

Our empirical findings demonstrate the effectiveness of the proposed compression and decompression technique, achieving compression ratios of $M/N \approx 0.019(=15/751)$ and $M/N \approx 0.027(=20/751)$ in Figures 2 and 3, respectively. The decompressed traces in panels (a1)-(a3) of Figures 2 and 3 closely align with the original digital-ink data, illustrating that the DP-based pseudo-input selection effectively preserves the intrinsic characteristics of handwriting (i.e. the original digital-ink data). In contrast, the uniform pseudo-input placement in panels (b1)-(b3) of Figures 2 and 3 resulted in slightly larger errors, particularly in regions with nonlinear or complex features. Generally, the precision was observed to augment with increasing M. However, with the error function used here, i.e. (27), pseudo-inputs may not be appropriately selected in linear sections, so there might be slightly larger errors in the decompressed areas, such as some part of the letter "h" as shown in Figures 2 and 3 (a1). Therefore, a different definition of the error function might be needed.



(a1) XY-plane with optimal selection (b1) XY-plane with equal interval selection

(a2) $X(t)$ with optimal selection (b2) $X(t)$ with equal interval selection

(a3) $Y(t)$ with optimal selection (b3) $Y(t)$ with equal interval selection

Figure 2: Compression and decompression results for the handwritten "hello" using the proposed method with $M = 15$.

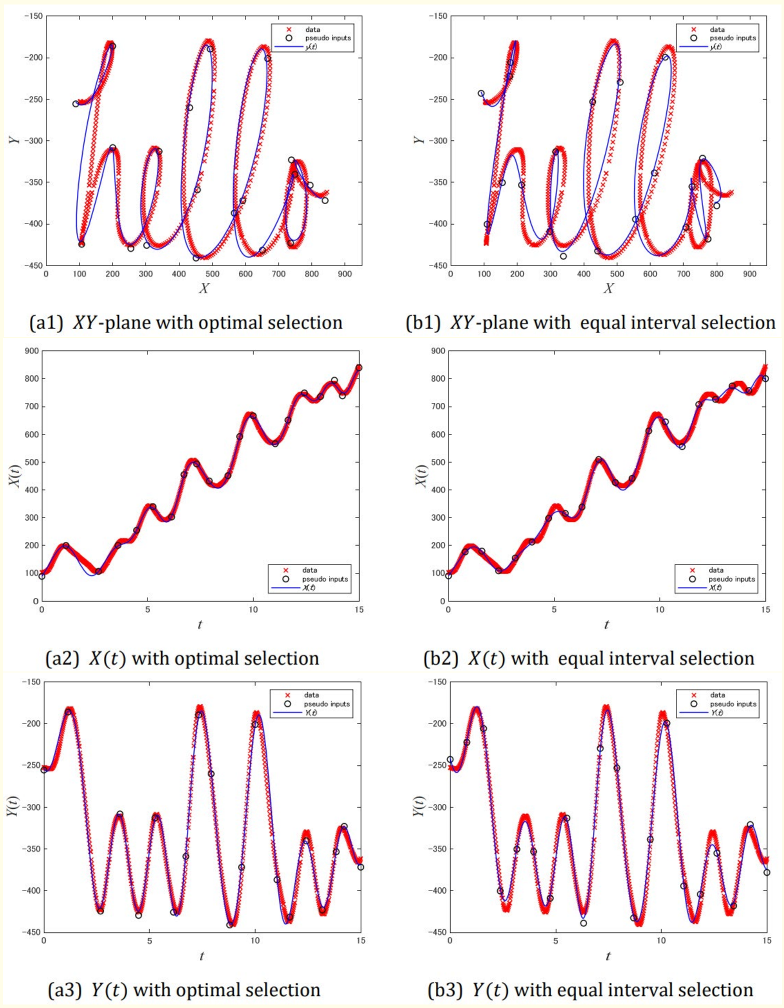


Figure 3: Compression and decompression results for the handwritten ”hello” using the proposed method with $M = 20$.

Conclusion

In this paper, we proposed a novel method for compressing and decompressing handwritten digital-ink data using sparse Gaussian process regression (GPR) approach. The method effectively reduces data size while preserving the intrinsic characteristics of the original digital-ink. In particular, we integrated a Dynamic Programming (DP) strategy to optimize the selection of pseudo-inputs in the sparse GPR, which significantly improved both the efficiency of data compression and the accuracy of data reconstruction. Through experimental studies, we demonstrated that compression ratios of approximately 0.019 (when using $M = 15$ pseudo-inputs) and 0.027 (when using $M = 20$ pseudo-inputs) were achieved with datasets containing 751 data points.

While the proposed method showed promising results in specific scenarios, further investigation is necessary to expand its applicability and robustness. Future work should explore the following key directions:

Extending applicability across diverse datasets and writing styles

Testing on a wider range of languages and handwriting patterns is essential to ensure the method's generalizability and practical utility.

Redefining or extending the error function and refining pseudo-input selection

To further improve pseudo-input selection and reduce reconstruction errors, especially in linear or highly variable regions, the error function may be refined by incorporating region-specific or adaptive criteria. For example, dynamically weighting areas with high curvature or complex strokes may help prioritize regions prone to reconstruction errors. By combining this with the current Dynamic Programming approach, pseudo-input selection can better adapt to diverse handwriting characteristics, enhancing both compression and reconstruction accuracy.

References

1. G Kurtenbach. "Pen-based Computing, XRDS: Crossroads". The ACM Magazine for Students-The Future of Interaction 16.4 (2010): 14-20.
2. V Mazalov and S Watt. "Digital Ink Compression via Functional Approximation". Proc. 12th Int. Conf. Frontiers in Handwriting Recognition, Kolkata, India (2010): 688-694.
3. R Hu and S Watt. "Optimization of Point Selection on Digital Ink Curves". Proc. 13th International Conference on Frontiers in Handwriting Recognition, Bari, Italy (2012): 525-530.
4. Y Mieno, H Fujioka and H Kano. "Data Compression of Digital-Ink with Pen-Slips Using Multi-level L1 Smoothing Splines". Proc. 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong (2015): 1787-1792.
5. R Tibshirani. "Regression Shrinkage and Selection via the Lasso". J. of the Royal Statistical Society Series B 58.1 (1996): 267-288.
6. SS Chen, DL Donoho and MA Saunders. "Atomic Decomposition by Basis Pursuit". SIAM Review 43.1 (2001): 129-159.
7. M Aharon, M Elad and A Bruckstein. "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations". IEEE Trans. Image Process 54-11 (2006): 4311-4322.
8. CM Bishop. "Pattern Recognition and Machine Learning". Springer (2006).
9. E. Snelson and Z. Ghahramani. "Sparse Gaussian Process using Pseudo-inputs". Neural Information Processing Systems 13. MIT Press (2006).
10. GH Golub and CF Van Loan. "Matrix Computations: Johns Hopkins Studies in Mathematical Sciences". Johns Hopkins University Press. 3rd edition (1996).
11. TA Davis. "Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms, Series Number 2)". Society for Industrial and Applied Mathematics (2006).