Compression of Digital Ink

Vadim Mazalov and Stephen M. Watt

Ontario Research Center for Computer Algebra
Department of Computer Science
University of Western Ontario

Joint Lab Meeting, University of Western Ontario
October 8, 2010

Western ORCCA

Introduction: Compression of Digital Curve

- Representing digital ink as points in a function space has proven useful for online recognition.
- This representation captures the shape of the ink traces with a small number of coefficients.
- This method is simple and yields high recognition results.
- Therefore, we deploy the same idea in compression of digital curve.
- A consequence of such representation is that it can be used in recognition almost directly.

Compression: Lossless vs. Lossy

- Lossless compression of digital ink is not a meaningful concept as each ink capture device has a resolution limit and sampling accuracy.
- Therefore, as long as the reconstructed curve lies within these limits, lossy and lossless compression are equivalent.
- As applied in handwriting recognition, lossless compression has no benefit – small perturbations in strokes give symbols that a human reader would recognize as the same.

Other Ink Compression Method

The most popular algorithm, deployed in the Microsoft ISF standard is compression based on the second differences

- The algorithm computes the second order differences of data items in each data channel.
- The sequence of second differences is proposed to have low variance and, therefore, be suitable for an entropy encoding algorithm.
- We show further on in the presentation that compression of our algorithm is considerably better.

Mapping the Interval of Approximation

- The interval of orthogonality in the definition of most of the classical orthogonal series is $\tau \in [-1, 1]$.
- If $f(\lambda)$ is defined on $[a, b]$, coefficients of the mapping function can be obtained by taking $\lambda = (b-a)\tau/2 + (a+b)/2$

$$\begin{aligned}\hat{c}_i &= \frac{1}{h_{ii}} \int_{-1}^{1} \hat{f}(\tau) B_i(\tau) w(\tau) d\tau \\ &= K_i \int_a^b f(\lambda) B_i(\frac{2\lambda - a - b}{b-a}) w(\frac{2\lambda - a - b}{b-a}) d\lambda\end{aligned}$$

where $K_i = \frac{2}{h_{ii}(b-a)}$. Then the d-th degree approximation of the original stroke can be found as

$$f(\lambda) \approx \sum_{i=0}^{d} \hat{c}_i B_i(\frac{2\lambda - a - b}{b-a}).$$

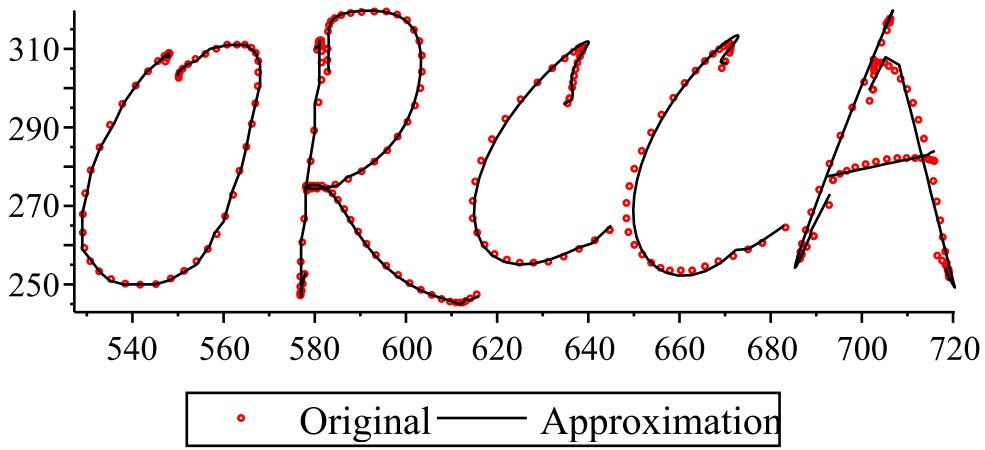
Bases for Approximation

We consider the following bases

- Chebyshev polynomials.
- Legendre Polynomials.
- Legendre-Sobolev polynomials.
- Fourier Series.

Problem Statement

We ask whether it is feasible to describe a stroke up to some given threshold of the maximal absolute error and root mean square error.



	O	R	C	C	A
Max. abs. error, %	1	2	3	4	5
RMSE, %	0.33	0.67	1	1.33	1.67

Overview: Compression

- Segment the stroke using one of the methods described below. Ensure the segments overlap by an amount at segmentation points.
- For each segment, compute the orthogonal series coefficients for each coordinate function (*e.g.* x, y, p)
- Deflate the stream of coefficients.

Overview: Restoration

- Inflate the coefficient stream to obtain the curve segments.
- Blend the curves on the overlaps to obtain the piecewise coordinate functions.
- Obtain traces by evaluating the coordinate functions with the desired sample frequency.

Parameterization of Coordinate Functions

We test the following, most popular in handwriting parameterization choices

- Time.
- Arc length.

Segmentation

We study the following approaches to segment a curve

- Fixed Degree Segmentation.
- Fixed Size Segmentation.
- Adaptive Segmentation.

Blending Segments

If we allow a large error threshold (e.g. 4%), then it becomes possible to notice naïve segmentation because we do not match derivatives at the segmentation points.

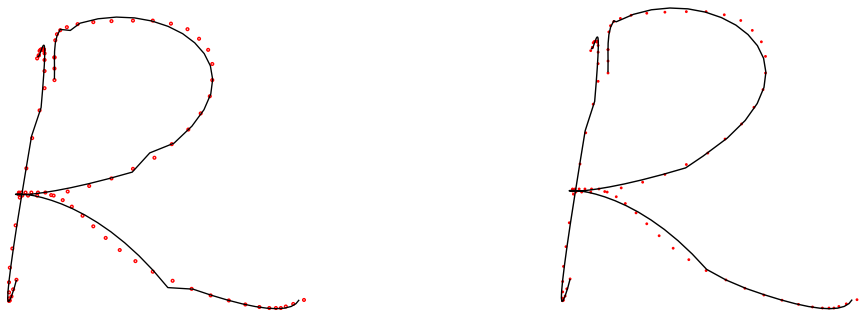


Figure: Example of blending

We propose to blend the transition from one piece to another by overlapping the segments slightly and transitioning linearly from one segment to the next on the overlap.

Blending Segments

The approximation is given in segments, f_j, and takes form

$$f(\lambda) = \sum_{j=1}^{N} W_j(\lambda) f_j(\lambda) \approx \sum_{j=1}^{N} W_j(\lambda) \sum_{i=0}^{d} c_{ij} P_i(\lambda)$$

with the weight function

$$W_j(\lambda) = \begin{cases} 0, & \lambda \leq \lambda_j - a \\ \frac{\lambda - (\lambda_j - a)}{a}, & \lambda_j - a < \lambda \leq \lambda_j \\ 1, & \lambda_j < \lambda \leq \lambda_{j+1} - a \\ \frac{-\lambda + \lambda_{j+1}}{a}, & \lambda_{j+1} - a < \lambda \leq \lambda_{j+1} \\ 0, & \lambda > \lambda_{j+1} \end{cases}$$

where a is a proportion of approximation pieces and λ_j are the segment transition points.

Representation

- We first look at textual representation of coefficients.
- Compressing textual representation of coefficients is relevant for standard XML representations.
- We store coefficients in UTF 8 format and define approximation packets as

$$\lambda_0; c^1_{00}, c^1_{01}, \ldots, c^1_{0d_{01}}; \ldots; c^N_{00}, c^N_{01}, \ldots, c^N_{0d_{0N}}$$
$$\lambda_1; c^1_{10}, c^1_{11}, \ldots, c^1_{1d_{11}}; \ldots; c^N_{10}, c^N_{11}, \ldots, c^N_{1d_{1N}}$$
$$\ldots$$

where λ_i is the initial parameterization value of piece i in the stream, N is the number of channels and d_{ij} is the degree of approximation of the piece i for j-th channel.

Experimental Setting

- We collected 108,094 points split in 1,389 strokes.
- Compressed size is reported in the experiments as the percentage of the original size of the whole dataset.

Size of Coefficients

- The question is how dependent the compression on the size of fractional part of coefficients.

- The result of the experiment for Chebyshev polynomials for the fixed degree method for different fractional sizes for the max. error of 3%.

F\D	3	5	7	9	11	13	15
0	2.62	2.49	2.53	2.79	3.05	3.31	3.59
1	3.91	3.69	3.62	3.70	3.69	3.70	3.64
2	5.36	5.18	5.10	5.29	5.24	5.27	5.21
3	6.82	6.65	6.58	6.87	6.81	6.84	6.80
4	8.29	8.13	8.07	8.45	8.37	8.42	8.38

- We take F=1, D=7. It allows to obtain the smallest compression for smaller error bounds.

Size of Coefficients: Other Error Thresholds and Bases

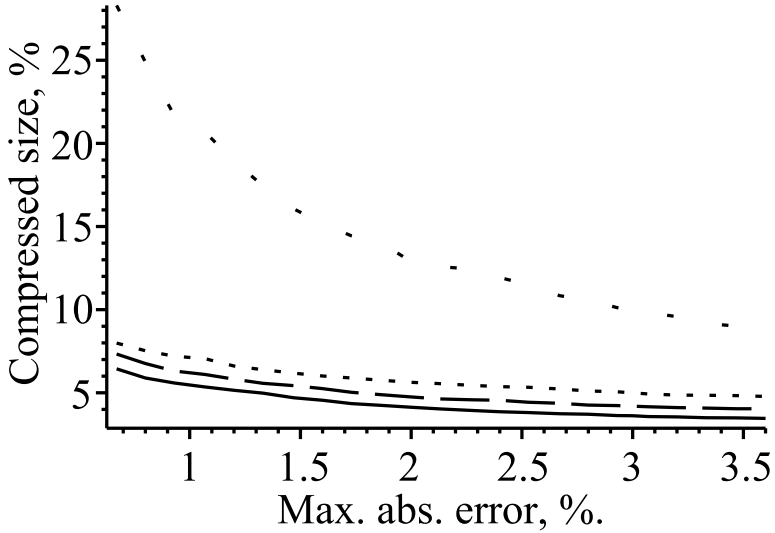


Figure: Compressed size for different values of error for Chebyshev (solid), Legendre (dash), Legendre-Sobolev (dot) and Fourier (space dot): coefficients with 1 digit fractional part

Parameterization by Time vs Arc Length

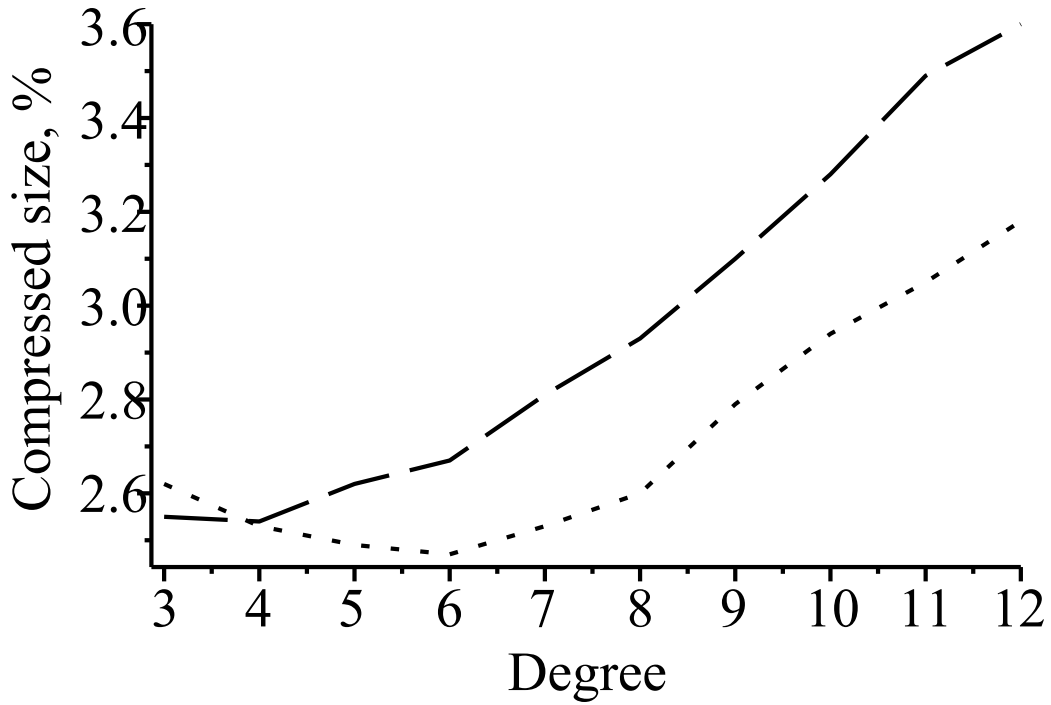


Figure: Compression for parameterization by time (dot) vs. arc length (dash) for different degrees of approximation, fractional part of size 0, Chebyshev polynomials for the max. error of 3%.

Fixed Length Approximation

"-" denotes the case, when an interval exists, that can not be approximated even with the orthogonal series of degree 20.

F\L	10	20	30	40	50	60
0	4.67	–	–	–	–	–
1	6.79	5.01	4.29	4.09	3.87	–
2	9.10	6.81	5.94	5.74	5.45	–
3	11.22	8.63	7.59	7.37	7.04	–
4	13.43	10.44	9.23	9.01	8.63	–

Table: Compressed size (%) by length of intervals (L) and fractional size of coefficients (F) for Chebyshev polynomials and max. error of 3%, fixed length method

Fixing the Size of Coefficients

We fix the number of digits in coefficients.

We allow the first coefficient to be twice of the size of coefficients of higher degree.

S\D	3	5	7	9	11	13	15
2	3.44	3.1	3.00	3.15	3.20	3.19	3.35
3	4.78	4.56	4.48	4.71	4.71	4.76	4.76
4	5.97	5.92	5.96	6.30	6.26	6.32	6.32
5	7.49	7.39	7.46	7.88	7.82	7.90	7.92

Table: Compressed size (%) for different approximation degrees (D) and coefficient sizes (S) for Chebyshev polynomials with max. error of 3%, fixed degree method

Other Error Thresholds and Bases

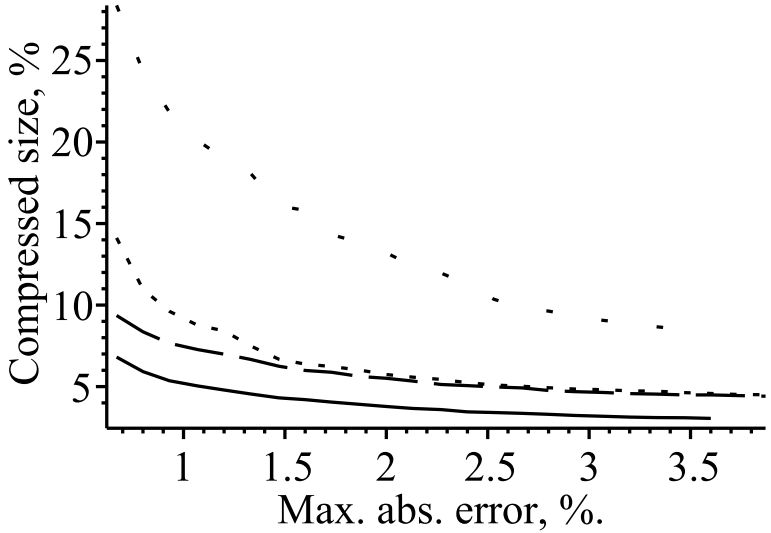


Figure: Compressed size for different values of error for Chebyshev (solid), Legendre (dash), Legendre-Sobolev (dot) for fixed size coefficients

Compression of Binary Traces

- We store the sequence of approximation coefficients compactly in an exponential format as ab where a and b are two's complement binary integers, standing for significand and a power of 10 respectively.
- We fix the size of b to 3 bits and change only the size of a.
- In the following experiments we use the adaptive segmentation scheme: we choose stroke-wise approximation parameters for each input channel separately.

Compression of Binary Traces

- Compression packets for each stroke i take the form

$$b_i; d_i; \lambda_1; c_{10}, c_{11}, \ldots, c_{1d_i}$$
$$\lambda_2; c_{20}, c_{21}, \ldots, c_{2d_i}$$
$$\ldots$$
$$\lambda_D$$

where b_i is the number of bits, d_i degree, λ_j initial value of parameterization of piece j and $c_{j0}, c_{j1}, \ldots, c_{jd_i}$ are coefficients.

Compression of Binary Traces: Results

B\E,%	0.0	0.6	1.1	1.5	2.0	2.5	3.1	3.5
△2	23.35	–	–	–	–	–	–	–
C	–	7.50	6.22	5.93	5.26	5.14	4.87	4.65
L	–	9.22	6.97	6.32	5.64	5.25	5.20	5.04
L-S	–	12.64	11.21	10.19	8.67	8.55	8.26	7.51

(a) binary coefficients

B\E,%	0.0	0.6	1.1	1.5	2.0	2.5	3.1	3.5
△2	8.64	–	–	–	–	–	–	–
C	–	3.07	2.61	2.31	2.05	1.90	1.80	1.72
L	–	3.41	2.86	2.53	2.26	2.08	2.00	1.91
L-S	–	9.36	7.27	6.25	5.51	4.98	4.64	4.49

(b) binary coefficients deflated

Table: Compressed size (%) for different errors (E) for representing coefficients in binary format for the second differences method (△2) and the following bases (B): Chebyshev (C), Legendre (L) and Legendre-Sobolev (L-S)

Compression of Binary Traces: Results

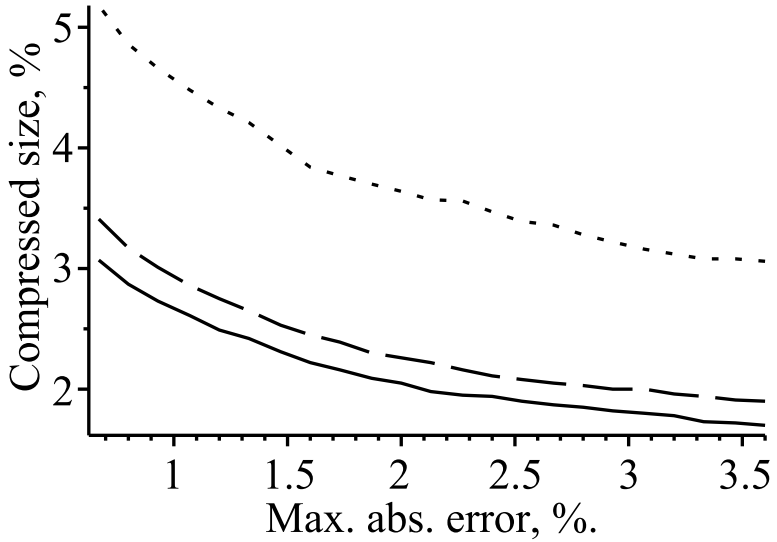


Figure: Compressed size for different values of error for Chebyshev (solid), Legendre (dash), Legendre-Sobolev (dot) for binary representation

Second Differences Method

- The second differences method yields high compression for low-resolution devices and vice versa, assuming that sampling rate remains the same.
- A stroke is represented by the values of the first two points and a sequence of second differences, since $x_{i\triangle 2} = x_i - 2x_{i-1} + x_{i-2}$.
- We store these values as binary numbers of fixed size, similar to the way, described previously.

Comparison with Second Differences

- Compression with Chebyshev polynomials for 1% maximum error yields 2.6% compressed size.
- For 2.5% (sampling error of the device) it yields 1.9% size.
- Maximum error $< 2.5\%$ is indistinguishable by a human and such compression can be accepted as lossless for the most of applications in pen-based computing.
- Lossless compression with second differences yields 8.64%.

Conclusion

We have developed several compression techniques and contributed with investigation of

- Parameterization by time vs arc length in approximation of digital strokes.
- Segmentation approaches: Fixed Degree, Fixed Length and Adaptive Segmentation.
- Blending segments for high error approximation.
- Ink packets format for representing characters in UTF-8 format.
- Ink packets format for representing characters in binary format.