Wang, J, et al. 2019. Efficient Stratified Sampling Graphing Method for Mass Data. *Data Science Journal,* 18: 56, pp. 1–9. DOI: https://doi.org/10.5334/dsj-2019-056

RESEARCH PAPER

Efficient Stratified Sampling Graphing Method for Mass Data

Jianjun Wang[1,2], Yingang Zhao[3], Jun Chen[4], Suqing Zhang[5], Xudong Zhao[5] and Yufei He[5]

[1] Lanzhou Geophysical National Observation and Research Station, Lanzhou, CN

[2] Gansu Provincial Earthquake Administration, Lanzhou, CN

[3] Anqiu Earthquake Station, Weifang, CN

[4] Anhui Earthquake Agency, Hefei, CN

[5] Institute of Geophysics, China Earthquake Administration, Beijing, CN

Corresponding author: Yingang Zhao (40857347@qq.com)

Sequentially linking data during polyline graphing of mass data (millions of points or more) generally results in poor graphing efficiency. Numerous curves are buried behind each pixel and cannot be displayed due to resolution limits of the width of the X-axis. Herein, a new efficient stratified sampling graphing method is proposed. The test results demonstrated that: (1) The full dataset is divided into *2X* subsets, where *X* is the width of the X-axis in pixels, and the maximum and minimum values of the data in each subset are respectively calculated and linked in order of appearance. This method yields *4X* sampled data graphs that are highly consistent with the full dataset graphs. (2) When the dataset is divided into *2X, 4X, 6X, 8X*, or more subsets (progressively increasing by even multiples), the similarity gradually increases. The average similarities can reach approximately 99.24% and 99.93% in the *2X* and *50X* subsets, respectively. We think that *2X* is the optimal subset allocation, which can achieve a high similarity, but also achieve the fastest sampling speed. (3) Compared with the speed of full dataset graphing, the overall speed of the "single-thread sampling + graphing" is increased by approximately 70 times, and that of the "threadPool sampling + graphing" was enhanced by approximately 200 times. The method employs the minimum amount of sampled data to obtain the full dataset graph that users expect to see, thereby significantly improving graphing speeds of mass data.

Keywords: mass data; polyline graphing; sampling graphing; stratified sampling; image similarity

Introduction

High sampling instruments are increasingly applied in scientific observation, producing ever-increasing amounts of data. For example, numerous types of time-domain instruments are used in seismographic observation, including continuous seismic waveform recorders at a sampling rate of 100 Hz and ACF-4M electromagnetic detectors at sampling rates of 1600 Hz and 3200 Hz. Users of such instruments often require a visual time-domain polyline graph of the collected data. The need to link these data sequentially slows graphing speeds down, hindering both software developers and users (Li Y et al, 2014; Zhou Y et al, 2010).

For example, there are three channels in a 100 Hz continuous seismic waveform instrument, and each channel generates 8,640,000 data points daily. The data acquired in one day (99 MB) requires approximately 46 s to graph in NET, which has been regarded as the fastest graphing after that obtained via optimization using common drawing technologies, such as double buffer graphing (Han Li-na et al, 2006), GraphicsPath graphing in NET (Zhou Y et al 2010), and threadPool's multicurve parallel graphing (Rodrigues C, 2014; Aulbach M, 2014). In China, there are thousands instruments that record data continuously and daily. A provincial agency must often compare, analyze, and process data generated from dozens of instruments on a single graphical interface. Currently, data processing must be performed at least 24 times daily as the data processing range is reduced to one hour due to slow graphing speeds.

Regarding the optimization of graphing speed, Hu L-D et al, (2014) proposed optimizing graphic component storage allocation strategies to increase the execution speed of the CPU instructions when performing hardware acceleration. Chou Z-D (2013) proposed the optimization of the local erase and redraw process by using double buffering, multithread, caching bitmap, hierarchical, timing, and delay refresh drawing mechanisms. Jiang J-P et al, (2006) proposed a period line-drawing algorithm in which more than one point is drawn by one calculation during the drawing of period lines, which greatly improves the line-drawing speed.

The slow graphing speed observed for mass data (millions of points or more) is mainly the result of the volumes of data, as a significant amount of time spent connecting lines. Current graphing optimization technologies are incapable of thoroughly solving this problem; thus, sampling-based graphing methods are required. Generally, a computer transverse resolution comprises approximately 2000 pixels. When the number of points to graph exceeds 2000 pixels, numerous curves are hidden behind each pixel and cannot be displayed. A key component of this technique is how to sampling from mass data and ensure that the sampled graphs are highly consistent with the full dataset graphs.

Various types of sampling methods have been proposed based on various requirements, such as greedy sampling (Chamon L et al, 2017), Monte Carlo sampling (Guo H et al, 2004), stratified sampling (Lu Y-G et al, 2017; Parsons V, 2017; Wessing S, 2017; Charles T, 2006), and density biased sampling (Palmer C et al, 2000). However, few studies have proposed an efficient sampling method for polyline graphing, although it has extensive applications in the fields of scientific research and data analysis. It can directly reflect changes in data patterns. To greatly improve the polyline graphing speed of mass data, a new efficient stratified sampling graphing algorithm is proposed in this paper. Based on the width of the X-axis in pixels, the full dataset is divided into *2X* subsets. The maximum and minimum values of all data points in each subset are calculated and linked in order of appearance. Therefore, *4X* sampled data graphs can be obtained.

To evaluate the consistency between the sampled and full dataset graphs, an image similarity algorithm was employed. Image similarity is of great significance in fields such as image recognition, face recognition, image tracking, and image search engines. The most common similarity calculation methods include pixel contrast, color histogram, and block segmentation (Zhu S, 2018; Li Q et al, 2017; Zhou C-M et al, 2016). Typically, two N-dimensional eigenvector is extracted from two images, such as $M = (m_1, m_2 \ldots, m_n)$ and $T = (t_1, t_2 \ldots, t_n)$. The distance between the two eigenvectors is calculated. The smaller the distance, the greater the similarity, and the larger the distance, the smaller the similarity. The Euclidean distance (Zhu S, 2018; Zhou C-M et al, 2016) is the most common distance calculation formula, which is used to measure the absolute distance between the points of two eigenvectors in a multidimensional space. This is an accurate method for dense and continuous data.

Herein, the implementation of an efficient stratified sampling graphing method is detailed. The graphing speeds of single-thread and threadPool sampling methods are compared and analyzed. Additionally, the similarity between sampled graphs of *nX* subsets and the full dataset graph is analyzed. The concept of gradual scaling control, which is suitable for sampled graphs, is also proposed.

1 Test data and research methods

1.1 Test data

The test data used in this study are obtained from 100 Hz continuous seismic waveforms. Each instrument has three channels: north–south, east–west, and vertical. Each channel generates 8,640,000 data points per day.

1.2 Efficient stratified sampling graphing method

To implement efficient stratified sampling, the width of the X-axis in pixels is used to divide the full dataset into 2*X* subsets (every two subsets correspond to a pixel). The maximum and minimum values of each subset are respectively calculated and linked in order of appearance (**Figure 1**). As an example, for a graph of 8,640,000 seismic waveform data points, if the width of the X-axis is 1000 pixels, the full dataset is divided into 2000 (2 × 1000) subsets, and the number of data points in each subset is 8,640,000/2000 = 4320. In the full dataset graphing process, although 8640 (4320 × 2) points in every two subsets are linked to each other, their connecting lines are fully projected onto the same pixel of the X-axis. Therefore, the connecting lines of four maximum and minimum values from these two subsets can be used to replace 8640 connecting lines.

Regardless of the total amount of data, the total amount of final sampled data is fixed to 4*X*, which depends on the number of pixels *X* in the width of the X-axis. If the total number of data points is below 4X, sampling will not be performed. In this case, the full dataset can be used directly to graph.

Efficient stratified sampling graphing includes the following steps (**Figure 2**):

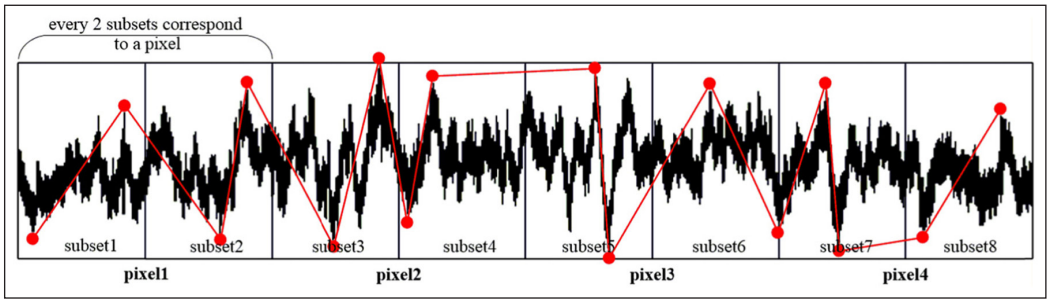

Figure 1: Example of efficient stratified sampling graphing. The black line represents a full dataset that is divided into eight subsets. The red line represents the sampled graph that links the maximum and minimum values (red dots) of each subset in their order of appearance.

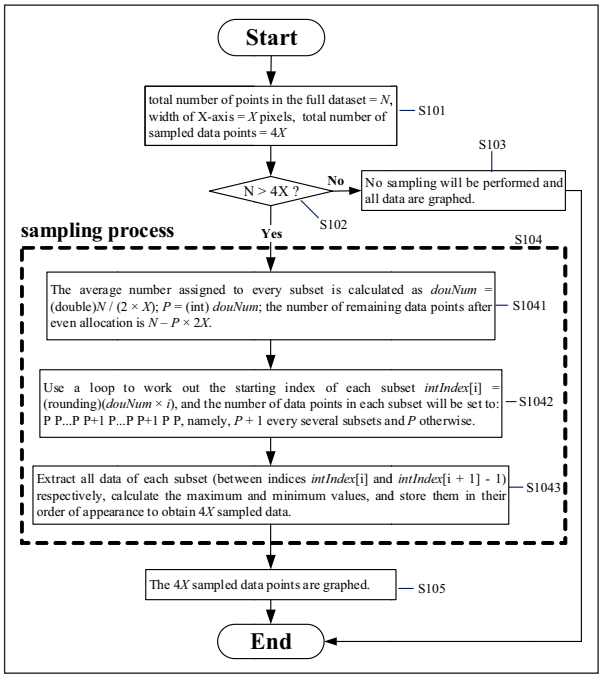

Figure 2: Flow chart of the efficient stratified sampling graphing method.

S101: Total number of points in the full dataset = N, width of the X-axis = X pixels, total number of sampled data points = $4X$.
S102: $N > 4X$?
S103: If no, sampling is not performed and all data are graphed.
S104: If yes, sampling is performed and $4X$ sampled data points are extracted.
S105: The $4X$ sampled data points are graphed.

The efficient stratified sampling method described in Step S104 includes the following steps (**Figure 2**):

Step S1041: The average number assigned to every subset is calculated as *douNum* = (double) $N/(2 \times X)$, where *douNum* is usually a decimal (N and $2X$ are not always divisible), and should be set to the *double* type to ensure sufficient accuracy, otherwise the last value of *intIndex* in Step S1042 may not equal N. Morreover, P = (int)*douNum*, where P is the minimum number of points assigned to each subset, and the number of remaining data points after even allocation is $N–P \times 2X$.
Step S1042: Use a loop to determine the starting index of each subset *intIndex*[i] = (rounding) (*douNum* × *i*), where $i = 0...2X$, *intIndex*[i] = 0...N (the last value must be N), and the forward and backward interval of each index in *intIndex* (or the number of data points in each subset) is set to P P...P P + 1 P...P P + 1 P P, namely, P + 1 every several subsets and P otherwise. Therefore, the number of remaining data points can be evenly distributed to the X-axis.

Step 1043: Extract all data of each subset (between indices *intIndex*[i] and *intIndex*[i + 1] – 1) respectively, calculate the maximum and minimum values, and store them in their order of appearance to obtain 4*X* sampled data.

1.3 Similarity testing method

The sampled and full dataset graphs are drawn in an area with the same pixel width W and height H. **Figure 3** shows the magnified polyline graph, which is composed of W straight lines perpendicular to the X-axis. Using 0 and 1 to represent white and black, respectively, a zero-one matrix of H rows and W columns can be obtained. Two eigenvectors are extracted from the two matrices, and the Euclidean distance between them can represent the similarity between the sampled and full dataset graphs. In this study, the following three methods are employed to test the similarity.

(1) **Pixel contrast method**, all pixels of the two matrices are compared and the sum of the pixels with the same color value *N* is calculated to determine the similarity, i.e., *N/(W* × *H)*.

(2) **Curve area method**, two W-dimensional eigenvectors can be obtained by calculating the sum of each column of two matrices. It represents the total number of black pixels per column, and it can also be understood as the area of the curve projected onto the X-axis.

(3) **Envelope line method**, the upper envelope line and the lower envelope line of the curve can be obtained by looking for the Y coordinates of the first 1 (black) from top to bottom and from bottom to top, respectively, in each column of the matrix. One 2W-dimensional eigenvector can be obtained by connecting the two envelopes together, and it contains the accurate position of the curve on the Y-axis. Thus, two 2W-dimensional eigenvectors can be obtained from the two matrices.

The Euclidean distance and similarities of the curve area method and the envelope line method are calculated using equations (1), (2), and (3), respectively. $H\sqrt{W}$ and $H\sqrt{2\times W}$ denote the maximum distance between the two eigenvectors, in this case, the Euclidean distance of each column of the two eigenvectors is all *H*.

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n}\left(x_i - y_i\right)^2} \tag{1}$$

$$Similarity = 1 - \frac{\sqrt{\sum_{i=1}^{w}\left(x_i - y_i\right)^2}}{H\sqrt{\mathrm{W}}} \tag{2}$$

$$Similarity = 1 - \frac{\sqrt{\sum_{i=1}^{2w}\left(x_i - y_i\right)^2}}{H\sqrt{2\times W}} \tag{3}$$

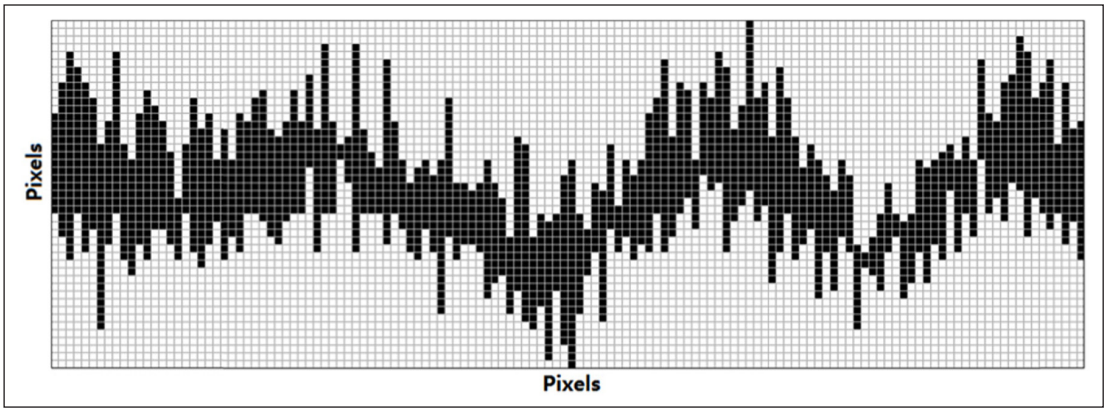

Figure 3: Magnified polyline graph.

Generally, the pixel contrast method is approximate, and only the total number of pixels of the same color is compared. The curve area method compares the curve area of each pixel on the X-axis but does not contain the position information of the curve on the Y-axis. The envelope line method compares the accurate position of the curve on the Y-axis, and its results are more accurate.

2. Test result

2.1 Comparison of the sampled and full dataset graphs

The full datasets comprise 8,640,000 points, and comparisons of the full dataset and sampled graphs are shown in **Figure 4**. There are partial minor differences between the full dataset (**Figure 4a**) and sampled (**Figure 4b**, *2X* subsets) graphs, though they are difficult to distinguish. **Figure 4c** shows a distortion when the full data are divided into 1X subsets.

2.2 Similarity between the sampled and full dataset graphs

Figure 5 (a), (b), and **(c)** present the similarity of *nX* subsets for sampling in the pixel contrast, curve area, and envelope line methods, respectively. The similarity of the three methods are consistent for 6,000, 60,000, 120,000, 600,000, 2,000,000, and 8,640,000 points. **Figure 5d** and **Table 1** show the average simi-

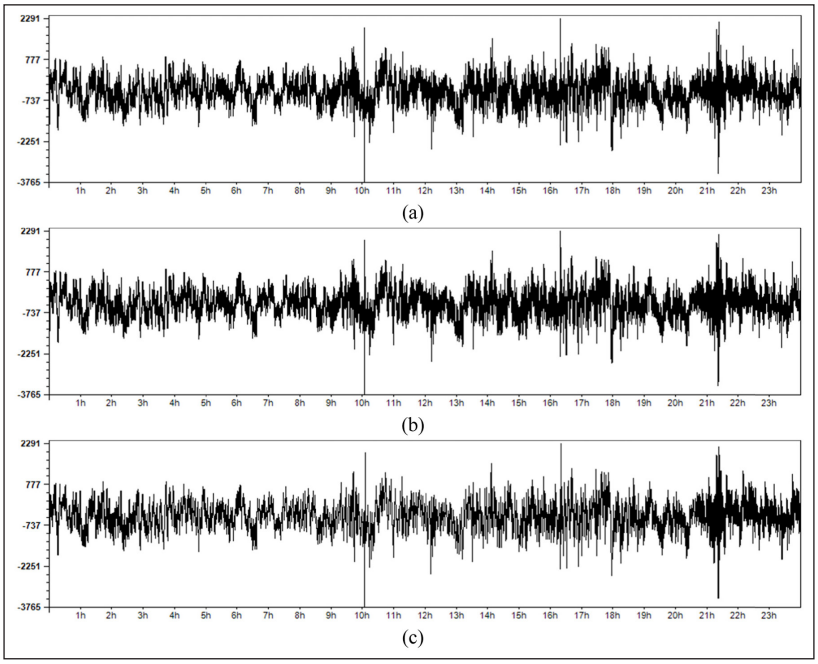

Figure 4: Comparison of full dataset graphs **(a)** and sampled graphs **(b, c)** when the full data are divided into *2X* (b) and *1X* (c) subsets in 8,640,000 points.

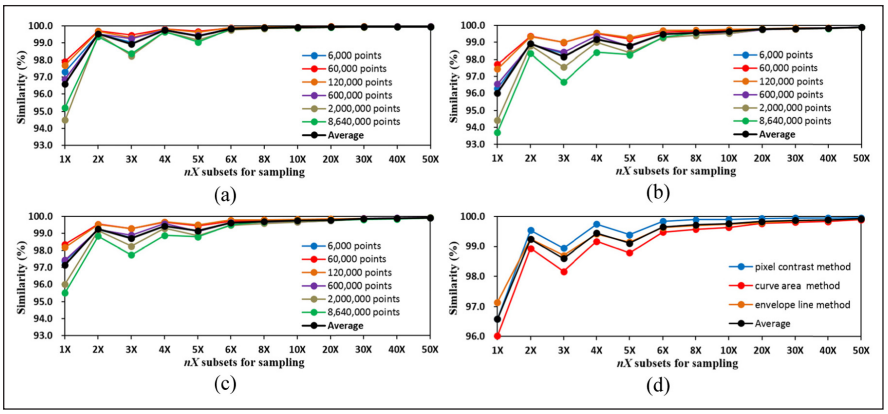

Figure 5: Similarity of *nX* subsets for sampling with **(a)** the pixel contrast method, **(b)** curve area method, and **(c)** envelope line method. **(d)** shows the average similarity of 6,000, 60,000, 120,000, 600,000, 2,000,000, and 8,640,000 points.

Table 1: Average similarity of the sampled and full dataset graphs.

Method	Average similarity (%) of *nX* subsets for sampling											
	1*X*	**2*X***	**3*X***	**4*X***	**5*X***	**6*X***	**8*X***	**10*X***	**20*X***	**30*X***	**40*X***	**50*X***
Pixel contrast method	96.58	99.54	98.94	99.75	99.39	99.84	99.89	99.91	99.94	99.95	99.95	99.96
Curve area method	96.02	98.93	98.16	99.17	98.78	99.47	99.58	99.64	99.77	99.81	99.83	99.89
Envelope line method	97.14	99.25	98.70	99.42	99.14	99.63	99.70	99.74	99.79	99.86	99.88	99.92
Average	96.58	99.24	98.60	99.44	99.11	99.65	99.72	99.76	99.83	99.87	99.89	99.93

Note: The width and height of the graphing window for the similarity test are 908 and 200 pixels, respectively. When the full dataset is divided into 1*X*, 3*X*, and 5*X* subsets, the similarity is lower; such items are marked in light gray.

Table 2: Sampling graphing speed test result.

Number of channels	Full dataset capacity (MB)	Full dataset graphing time (s)	Single-thread sampling + graphing				threadPool sampling + graphing			
			Sampling time (s)	Graphing time (s)	Total time (s)	Raising rate	Sampling time (s)	Graphing time (s)	Total time (s)	Raising rate
3	99	46	0.55	0.18	0.73	**63.0**	0.14	0.20	0.34	**135.3**
36	1186	546	6.43	1.34	7.77	**70.3**	1.06	1.25	2.31	**236.4**
72	2372	1072	13.04	2.69	15.73	**68.2**	1.98	2.51	4.49	**238.8**

Note: Graphing employs threadPool technology, with one thread per channel; Total time = sampling time + graphing time; Raising rate = Full dataset graphing time/total time.

larity. **Figure 5d** indicates that the pixel contrast method presents the highest similarity, followed by the envelope line method and curve area method. The similarity of the envelope line method should be more reliable as it compares the accurate position of the curve on the Y-axis. The average similarity of the three methods is used to evaluate the consistency between the sampled and full dataset graphs.

When the data are divided into *1X* subsets, the similarity is generally <96.58%. When the data are divided into *3X* or *5X* subsets, the similarity is slightly lower than when divided into *2X* or *4X* subsets. If the full dataset is divided into *2X*, *4X*, *6X*, *8X*, or more subsets (increasing by even multiples), the similarity gradually increases. When the full dataset is divided into *50X* subsets, the similarity reaches approximately 99.93%, at which point the sampled and full dataset graphs are nearly identical.

2.3 Sampling plus graphing speed test

The sampling process in **Figures 1** and **2** is fully operated in memory. It is essentially equivalent to a full dataset round-robin. Although the sampling speed is much faster than the full dataset graphing speed, it still decreases for larger datasets (millions of points or more). The sampling speed can be further optimized by threadPool, which allows to use multiple threads in parallel and sharply increases sampling speeds.

Continuous 100 Hz seismic waveform data are divided into 3, 36, and 72 channels. **Table 2** shows the test results of full dataset graphing, single-thread sampling + graphing, threadPool sampling + graphing (unit of time is seconds). The single-thread sampling time represents the vast majority of the total time. As the full dataset increases in size, sampling becomes slower. After threadPool sampling is applied, the sampling speed increases significantly. Compared with the full dataset graphing speed, the overall speed of "single-thread sampling + graphing" is approximately 70 times faster and "threadPool sampling + graphing" is approximately 200 times faster.

2.4 Concept of gradual scaling control

Users often gradually magnify curves to inspect more buried data, but the sampled data do not include the full dataset. When magnifying, the selected data range must be confirmed acorrding to the mouse positioning. The difficulty with it is to determine an accurate mapping between the mouse-positioned pixels and real data points. The concept of a gradual scaling control that is appropriate for sampled graphs is shown in **Figure 6**.

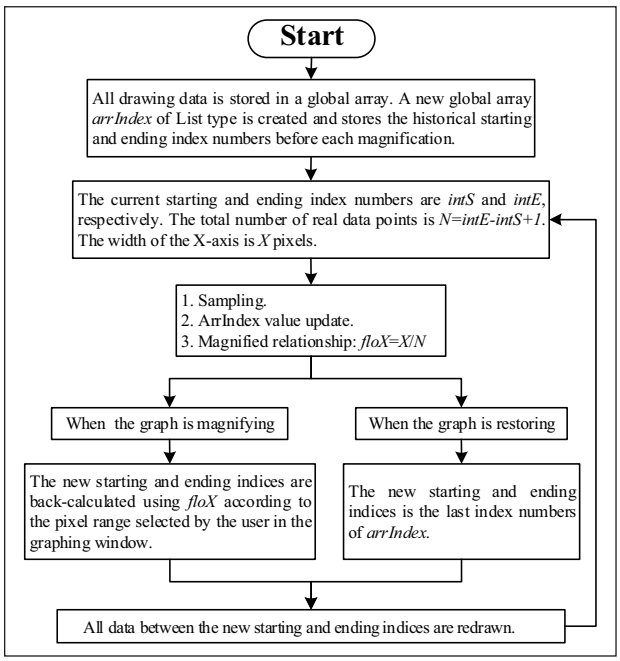

Figure 6: Concept of gradual scaling control appropriate for sampled graphs.

A new global array *arrIndex* of List type is created and stores the historical starting and ending index numbers before each magnification. The current starting and ending index numbers are *intS* and *intE*, respectively. The total number of real data points is $N = intE - intS + 1$. The width of the X-axis is *X* pixels.

Three steps are as followings. (1) Sampling: When $N > 4X$, sampling is performed for all data between *intS* and *intE*; otherwise, no sampling occurs. (2) *ArrIndex* value update: If the current graph is magnified, *intS* and *intE* are appended to the end of *arrIndex*. If the current graph is restored, the last index of *arrIndex* is removed. If the current graph is a full dataset graph without any magnification, *arrIndex* is emptied. (3) Magnified relationship: The average pixel width of each data point on the X-axis is calculated as $floX = X/N$.

The new starting and ending index numbers are back-calculated using *floX* according to the pixel range selected by the user in the graphing window when the graph is magnifying, or is the last index numbers of *arrIndex* when the graph is restoring. All data between the new starting and ending indices are redrawn.

3 Discussion and conclusion

3.1 Discussion

Figure 4 shows the partial and minor differences between the sampled (**Figure 4b**, *2X* subsets) and full dataset graphs (**Figure 4a**), which are often difficult to distinguish visually. These minor differences occur because the current allocation of each subset does not strictly correspond to each pixel on the display. For example, the maximum or minimum value of a subset appears exactly at the junction of subsets or pixels. The value that would be assigned to the previous subset is actually assigned to the next subset, resulting in minor and partial differences in the graphs. Graphs are only a reference tool and these slight differences are negligible for users. Further, these differences disappear after partial magnification (if *N* is not greater than *4X*, no sampling is performed and all data are graphed).

Distortion, as shown in **Figure 4c**, occurs when the full dataset is divided into *1X* subsets. As well as lower similarity in **Figure 5** when the dataset are divided into *3X* or *5X* subsets. Odd subsets will cause more partial differences between the sampled and full dataset graphs than even subsets.

The average number assigned to every subset (*douNum*) in Step S1041, as shown in **Figure 2**, is set to the double type as the total number of data points *N* and *2X* is not always divisible, allowing the even distribution of the remaining data to the X-axis. The minimum number (P) of each subset is increased by 1 every several subsets. The test results demonstrate that when the remaining data are allotted to these subsets in the head or tail (the number of the data points in these subsets is P + 1, and the rest is P), there is a deviation in the starting and ending index numbers back-calculated from the selected pixel range during graph magnification, and the magnified graph does not show the starting and ending area selected by users. However, this deviation does not occur again after an even distribution.

In the flow chart illustrating the efficient stratified sampling algorithm in **Figure 2**, the full dataset should be divided into *2X* subsets, which correspond to the smallest available number of subsets under this algorithm. At this time, the total number of sampled data points is *4X*, and on average, approximately four sampled data points are used to determine one pixel on the display. Thus, the average similarity between the sampled and full dataset graphs can reach at least 99.24%. When the dataset is divided into *2X*, *4X*, *6X*, *8X*, or more subsets (progressively increasing by even multiples), the similarity gradually increases, and when the full dataset is divided into *50X* subsets, the average similarities can reach approximately 99.93%. At this time, the sampled graph is almost identical to the full dataset graph. However, as the number of subsets increases, the sampling speed decreases. Therefore, we think that *2X* is the optimal subset allocation, which can achieve high similarity, but also achieve the fastest sampling speed. If the sampling graph requires higher similarity, the number of subsets can be increased to *50X* (or more).

The efficient stratified sampling graphing method proposed in this study has been applied to the Ground Pulsation Short-term Prediction and Real-time Tracking System, which had previously been developed. This system is currently applied in the Gansu, Qinghai, Sichuan, Yunnan, Tibet, and Xinjiang Agencies of China and will be further promoted and used in the national seismic system. The high graphing efficiency will have wide applicability to mass seismic waveform data.

3.2 Conclusions

To improve the poor polyline graphing speed of mass data in the time domain, the implementation of an efficient stratified sampling graphing method was detailed. The graphing speeds of single-thread and threadPool sampling methods were compared and the similarity between the sampled and full dataset graphs was analyzed when the full data are divided into *nX* subsets. The concept of gradual scaling control, which can be applied to the sampled graphs, was also proposed. The test results demonstrated the following points: (1) When the full dataset is divided into *2X* subsets based on the width of the X-axis in pixels, and the maximum and minimum values of all data in each subset are calculated and linked in order of appearance, *4X* sampled data graph can be obtained, which is highly consistent with the full dataset graph. (2) When the dataset is divided into *2X*, *4X*, *6X*, *8X*, or more subsets (progressively increasing by even multiples), the similarity gradually increases. The average similarities can reach approximately 99.24% and 99.93% in the *2X* and *50X* subsets, respectively. We think that *2X* is the optimal subset allocation, which can achieve high similarity, but also achieve the fastest sampling speed. (3) Compared with the speed of full dataset graphing, "single-thread sampling + graphing" is 70 times faster and "threadPool sampling + graphing" is 200 times faster.

The efficient stratified sampling graphing method employs the minimum amount of sampled data to obtain the full dataset graph that users expect to see, thereby greatly improving graphing speeds for mass data. This method offers high reference value for software development, graphing design concepts, data mining and analysis, etc. It is very simple and easy to extend to other languages. The sampling algorithm is also applicable to sequential polyline graphing outside the time domain.

However, the efficient stratified sampling graphing method proposed in this study is only suitable for polyline graphing with equal sampling intervals. Additionally, the sampled graphs are only highly consistent with the full dataset graphs and not identical. Therefore, our future research on computer graphics focus on improving sampling abilities and making sampled graphs identical.

Acknowledgements

We would like to thank the China Earthquake Networks Center for providing all the test data used in this study.

Funding Information

This work was jointly supported by the Spark Program Project for Seismic Science and Technology of the China Earthquake Administration (XH17038) and the Project of Earthquake Science Experimental Site in China (2019CSES0108).

Competing Interests

The authors have no competing interests to declare.

Author Information

Jianjun Wang, male, born in 1975, senior engineer, Master's degree, engaged in seismic electromagnetic observation method research and technical management.

Yingang Zhao, male, engineer, Bachelor's degree, engaged in seismic monitoring and seismic software research and development.

References

Aulbach, M, Fink, M and **Schuhmann, J,** et al. 2014. Drawing graphs within restricted area. *Computer Science*, 8871: 367–379. DOI: https://doi.org/10.1007/978-3-662-45803-7_31

Chamon, L and **Ribeiro, A.** 2017. Greedy sampling of graph signals. *IEEE Transactions on Signal Processing*. 66(1): 34–47. DOI: https://doi.org/10.1109/TSP.2017.2755586

Charles, T. 2006. Refinement strategies for stratified sampling methods. *Reliability Engineering & System Safety*, 91(10): 1257–1265. DOI: https://doi.org/10.1016/j.ress.2005.11.027

Chou, Z-D. 2013. Windows High-Speed Drawing Technology Research. *Applied Mechanics and Materials*, 441: 660–665. DOI: https://doi.org/10.4028/www.scientific.net/AMM.441.660

Guo, H, Hou, W-C and **Yan, F,** et al. 2004. A Monte Carlo sampling method for drawing representative samples from large databases. *International Conference on Database Systems for Advanced Applications.* Berlin, Heidelberg: Springer. DOI: https://doi.org/10.1109/SSDM.2004.1311239

Han, L-N and **Shi, H-S.** 2006. Dealing with flicker problem about GDI with double buffering. *Computer Engineering and Design*, 27(17): 3258–3260. DOI: https://doi.org/10.3969/j.issn.1000-7024.2006.17.048

Hu, L-D and **Zhang, J.** 2014. Research on embedded graphics system performance optimization based on GTK+. *Computer Engineering*, 40(1): 287–290. DOI: https://doi.org/10.3969/j.issn.1000-3428.2014.01.062

Jiang, J-P and **Zeng, L-Q.** 2006. Improving Line-drawing Speed with Line Period. *Computer & Modernization*, 11: 79–81. DOI: https://doi.org/10.3969/j.issn.1006-2475.2006.11.026

Li, Q, You, X and **Li, K,** et al. 2017. Geography Image Similarity Measurement Method Based on Adaptive Weighting of Similarity Matrix. *PR & AI*, 30(11): 1003–1011. DOI: https://doi.org/10.16451/j.cnki.issn1003-6059.201711005

Li, Y, Xu, J and **Wang, C-H.** 2014. Plotting images of volume data with high efficiency without blinking In Visual C++. *Software Development and Design*, 2: 19–20. DOI: https://doi.org/10.16184/j.cnki.comprg.2014.02.016

Lu, Y-G and **Huang, J-T.** 2017. Study on stratified random sampling based on R software. *Statistics & Decision*, 22: 36–39. DOI: https://doi.org/10.13546/j.cnki.tjyjc.2017.22.008

Palmer, C and **Faloutsos, C.** 2000. Density biased sampling: An improved method for data mining and clustering. *ACM*. DOI: https://doi.org/10.1145/342009.335384

Parsons, V. 2017. Stratified sampling. *Wiley StatsRef: Statistics Reference Online*. DOI: https://doi.org/10.1002/9781118445112.stat05999.pub2

Rodrigues, C. 2014. Supporting high-level, high-performance parallel programming with library driven optimization. Illinois: University of Illinois at Urbana-Champaign. http://hdl.handle.net/2142/49427.

Wessing, S. 2017. Experimental analysis of a novel stratified sampling algorithm for hypercubes. *Germany: Technische Universität Dortmund*. https://arXiv.org:1705.03809.

Zhou, C-M, Xue, D and **Guo, S-S,** et al. 2016. Improved Image Similarity Algorithm. *Computer Science*, 43(6): 72–76. DOI: https://doi.org/10.11896/j.issn.1002-137X.2016.6.015

Zhou, Y and **Zhang, L-C.** 2010. Some skills for improving drawing speed. *Value Engineering*, 6: 51. DOI: https://doi.org/10.14018/j.cnki.cn13-1085/n.2010.06.042

Zhu, S. 2018. Image Similarity Computation by Using Histogram Method. *Bulletin of Surverying and Mapping*, 501(12): 96–100. DOI: https://doi.org/10.13474/j.cnki.11-2246.2018.0391

How to cite this article: Wang, J, Zhao, Y, Chen, J, Zhang, S, Zhao, X and He, Y. 2019. Efficient Stratified Sampling Graphing Method for Mass Data. *Data Science Journal,* 18: 56, pp. 1–9. DOI: https://doi.org/10.5334/dsj-2019-056

Submitted: 16 July 2019 **Accepted:** 31 October 2019 **Published:** 13 November 2019

Copyright: © 2019 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See http://creativecommons.org/licenses/by/4.0/.

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS