Intelligent compression for synchrotron radiation source image

Shiyuan Fu[1,2,*], *Lu Wang*[1,2], *Yaodong Cheng*[1,2,3], *Gang Chen*[1]

[1]Institute of High Energy Physics, CAS, 100049 Beijing, China

[2]University of Chinese Academy of Sciences, 100049 Beijing, China

[3]Tianfu Cosmic Ray Research Center, Institute of High Energy Physics, Chinese Academy of Sciences, 610041 Chengdu, China

Abstract Synchrotron radiation sources (SRS) produce a huge amount of image data. This scientific data, which needs to be stored and transferred losslessly, will bring great pressure on storage and bandwidth. The SRS images have the characteristics of high frame rate and high resolution, and traditional image lossless compression methods can only save up to 30% in size. Focus on this problem, we propose a lossless compression method for SRS images based on deep learning. First, we use the difference algorithm to reduce the linear correlation within the image sequence. Then we propose a reversible truncated mapping method to reduce the range of the pixel value distribution. Thirdly, we train a deep learning model to learn the nonlinear relationship within the image sequence. Finally, we use the probability distribution predicted by the deep leaning model combined with arithmetic coding to fulfil lossless compression. Test result based on SRS images shows that our method can further decrease 20% of the data size compared to PNG, JPEG2000 and FLIF.

1 Background

1.1 High Energy Photon Source (HEPS)

[*] Corresponding author: fusy@ihep.ac.cn

The High Energy Photon Source (HEPS) under construction at Huairou, Beijing is one of the world's brightest fourth-generation synchrotron radiation facilities. It will offer an nm level spatial resolution, ps level time resolution, and MeV level energy resolution. The first-stage of HEPS will be equipped with 14 beamlines to users. The experiments will generate large amounts of data. The HEPS beamlines in the first-stage project are estimated to produce an average of 200TB raw data per day, with a peak value of 500TB per day [1]. The total amount of data generated in a year will be about 150PB, among them the SRS images generated by the hard X-ray imaging beamline account for the majority, which accounts for more than 90% of the total data generated by HEPS. Therefore, the hard X-ray image data requires the largest capacity of storage and bandwidth. The SRS images mentioned in this paper are all hard X-ray images.

The data produced by HEPS will not only increase continuously but also require long-term preservation. This situation brings big challenge to data storage and transmission. A simple expansion of storage capacity and bandwidth cannot solve the problem fundamentally, and it requires a lot of research funding. Data compression is one of the effective ways to reduce the amount of data. To ensure the potential scientific value in data, information cannot be lost during preservation and transmission, the compression and decompression must be lossless.

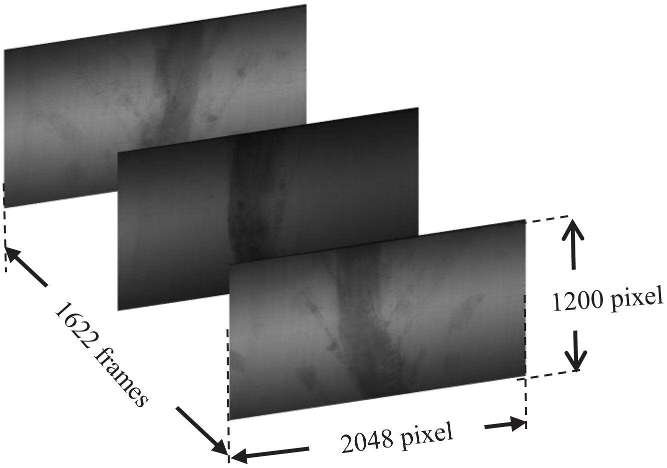


Figure 1, An example of SRS image sequence.

1.2 SRS Image characteristics

The imaging process of SRS images is projection-imaging sequences of different angles of one sample. The sample rotates a fixed minimum angle around its vertical central axis in each shooting gap, and the hard X-ray penetrates the sample and generates a projected image of the sample. The value of the SRS image pixel is proportional to the number of photons hitting on the sample. SRS images have the characteristics of high resolution, high frame rate and high contrast. An example is shown in Figure 1.

The pixel value range of images is 0~65535. The image size currently is 2k×2k, and it will gradually increase in the future. It is expected to reach 6k×6k next year and eventually reach 10k×10k or higher. The frame number of one image set is related to its resolution. At present, the frame number of one image dataset is about 2000 at a resolution of 2k×2k. With the increase of the resolution, frame number can reach tens of thousands. Therefore, each sample imaging will generate gigabytes or even terabytes of data.

2 Related work

The compression rate (CR) can indicate how much storage capacity is occupied after compression. CR equals the compressed size divide by the original size. The smaller the CR, the better the compression method and the less storage and bandwidth resources required.

Traditional lossless compression methods are generally simple, such as PNG [2], JPEG2000 [3], FLIF [4] and other mathematical methods. However, those methods cannot be optimized dynamically for different data type because of the fixed calculation process. More importantly, the existing methods can only save up to 30% in size in the case of SRS image, as shown in figure 2.

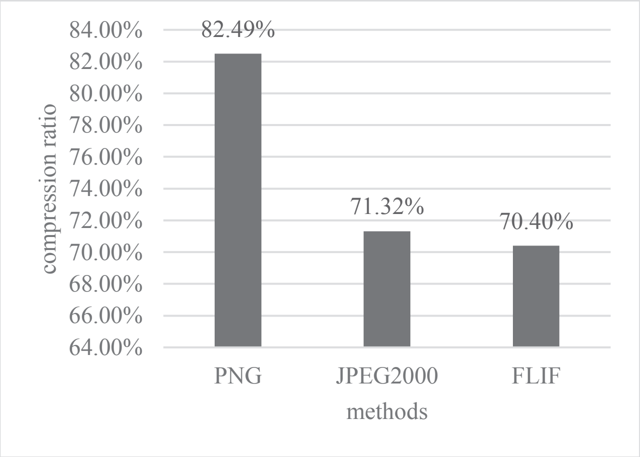


Figure 2, Compression ratio of different methods, test with SRS image dataset collected on Shanghai synchrotron radiation source.

In recent years, the algorithms of deep learning has been widely applied in many areas, such as speech recognition [5] and machine translation [6], while compression is also included. The Deepzip[7] uses an RNN model as the probability predictor and encodes data by arithmetic coding. The probability predictor predicts the probability distribution of the current data from the forward sequence and the distribution is used to compress current data by dynamic arithmetic coding.

However, this method can hardly reduce the size when directly applied to the SRS images. The bad result is due to the wide range of values each pixel can assume in different frames. Therefore, the pixel value of the frames has a wide distribution range. The probability distribution interval corresponding to each value is small, which cannot make full use of neural network and arithmetic coding, resulting in no reduction in data size.

Therefore, we propose a new method to reduce the distribution range of pixel value, and to make the same network structure work on SSR image as the probability predictor.

3 Implementation

3.1 Architecture

The method in this work is composed of the following four steps: mapping, modelling, predicting and arithmetic coding, as shown in figure 3. The image sequence is sorted by timestamp. First, it passes through the difference algorithm module to get I_t', and then get I'_{t_map} after mapping. The modelling step uses one image in the sequence and its previous k images as training data to train the deep learning model. After the model is built, the probability distribution will be obtained as the output. Then all the frames of one sample similar to that introduced in section 2.1 are compressed using the probability distribution combined with arithmetic coding. Finally, the compressed data is saved into the storage system. At the same time, the model parameters are also saved to ensure that there is no loss in the compression and decompression process.

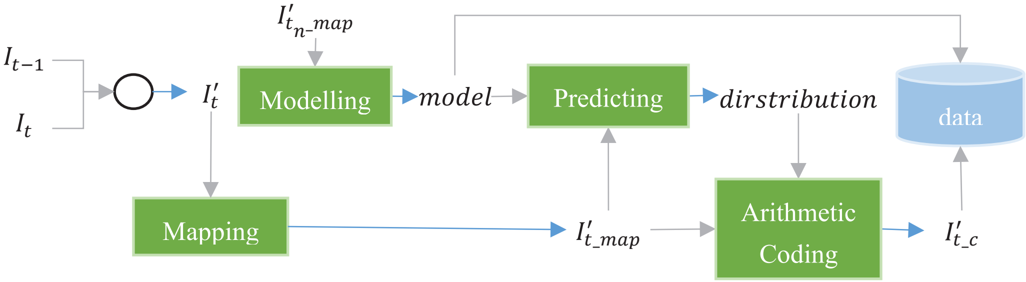


Figure 3, The architecture of our method, The gray arrow represents the input stream, the blue arrow represents the output stream.

3.2 Image Difference

Image difference technology is used to reduce the correlation between images or inside one image, and highlight different parts. As shown in figure 4, the first row is the original image sequence sorted by time from left to right, and the second row is the difference image sequence obtained by subtracting the pixel value of the corresponding spatial position of the previous image. The contour of the original image sample is blurred. At the same time, due to the image characteristics, the pixel that contains the sample information has a smaller value while the other part that does not contain the sample information has a larger value. After the difference, sample profile features are more obvious, and the pixel value that contains the sample information is increased. Since the rotation angle of the sample during the imaging of adjacent frames is extremely small and the imaging time is extremely short, the environmental noise of adjacent frames are relatively close to each other, so the difference can reduce the noise of the image.

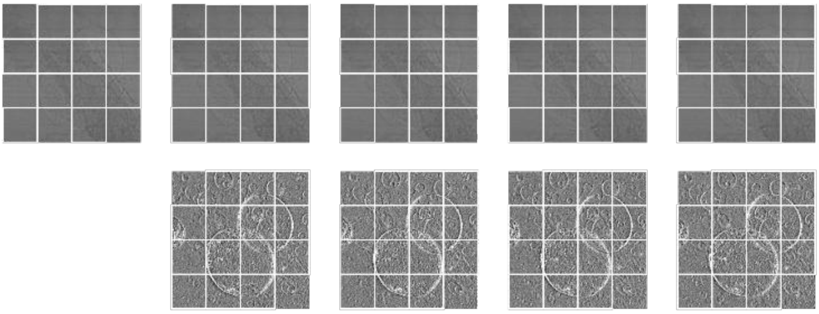
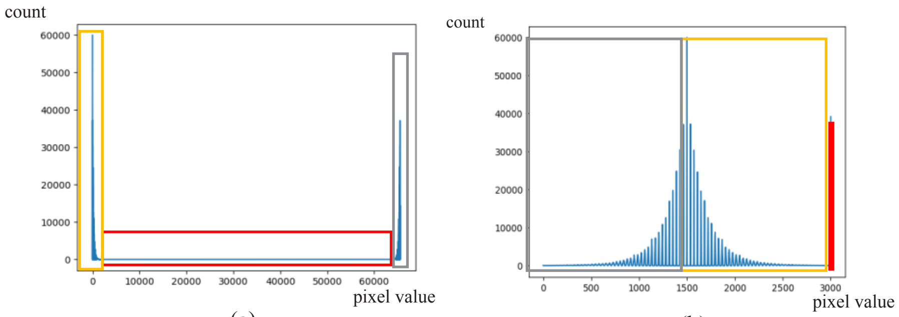


Figure 4, Comparison before and after difference. These images sorted by timestamps are small parts of the images in same timestamp shown in Figure 1.

3.3 Mapping

This method is used to narrow the range of values and needs to ensure that the process is reversible and no information is lost. As shown in Figure 5(a), the distribution of image pixel values is relatively more concentrated after difference, and most of the data is distributed in a relatively small range. Therefore, the data is divided into two parts. One part is the majority part contains the most data; the other part contains the residual data. Only the majority will be compressed. In Figure 5(b), the yellow part and the grey part are the ranges that contain most information, while the red part containing very few data occupies most of the distribution area. So, we can shift the yellow part and grey part to the right and map to a new range starting from zero. The red part data will be mapped to a fixed data, which may be the next number of the largest number of the new range. In this way, we can reduce the data distribution, and offset the data volume expansion caused by a small portion of uncompressed data.



(a) (b)

Figure 5, Correspondence of each part in the calculation process of truncated mapping

Therefore, we proposed truncated mapping to get a smaller data distribution range in exchange for saving a small amount of uncompressed data. Truncated mapping ignores the pixel values in the red range, which accounts for less than 2% of the total, ~~within 2%~~ to obtain a new data range, and maps the uncompressed part to the end of the new range plus one value, and the original value will be placed at the end of the compressed file in chronological order for lossless decompression.

3.4 Modelling

The arithmetic coding includes static arithmetic coding and dynamic arithmetic coding. The former is faster, and the latter has a lower compression rate. The main purpose of our method is to pursue a low compression rate, so a dynamic arithmetic coding strategy is adopted. Dynamic arithmetic coding needs to dynamically update the probability distribution of a certain length of data to be compressed. So the target of modelling is to provide such a probability distribution, and the length of the data to be compressed is 1.

3.4.1 Model architecture

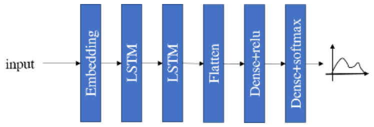


Figure 6, Example of model architecture

The modelling is used to train a model that learns the nonlinear relationship within the image sequence, so as to predict the probability distribution of the current pixel from the forward sequence. The frames here have undergone difference and mapping. Because the frames are arranged in chronological order, so we arrange the pixels at the same position of different frames into a sequence. The distribution probability of the current pixel value is predicted from the forward values. The probability corresponding to the true pixel value is what arithmetic coding needs, so as to achieve the purpose of lossless coding.

We use three models as the probability predictor: fully connected networks (FC), long short-term memory (LSTM) and gated recurrent units (GRU). The model architecture is as shown in Fig. 5.5, using LSTM as an example.

3.4.2 Input & output

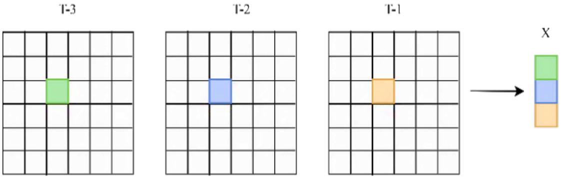


Figure 7, Model input example

Taking the probability distribution of a certain pixel value of the image at time T (the red pixel block at time T in Figure 7 as an example, the input is that the pixels with the same spatial position as the predicted pixel value in the first K images arranged in time order. The input sequence of Figure 7 is the case of K=3, where K is the length of the time series.

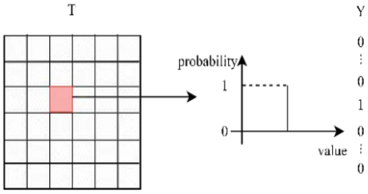


Figure 8, Model output example

The output is a sparse vector with the same length as that of the pixel value range. The value of each element in the vector represents the possibility if the pixel value equals the corresponding element index. The output example is shown in Figure 8. The element value is 1 only at the index corresponding to the predicted pixel value, while others are 0. The output of the model is the probability distribution of the predicted value in predicting.

4 Result

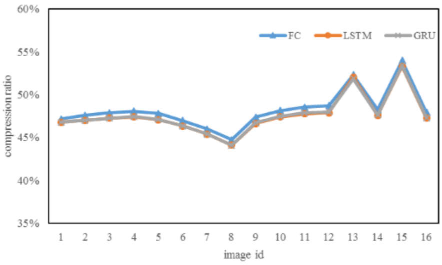


Figure 9, Result of different models

The way to build an independent model for every image will consume a lot of time and space. Take K=8 as an example, the size of different models are different, for example, the FC model is about 2.4MB, while the LSTM and GRU models are about 1.2MB. As mentioned before, the model parameters need to be saved as one part of the compressed image. The model size seriously affects or even cancels out the reduced size of the compressed image. Therefore, it is necessary to train a global model for one image sequence. Compared to Deepzip that building a model for each data, we only use one image to train a model for multi images because the image sequence obtained from one sample is similar to each other.

4.1 Dataset

Our dataset is shown in figure 1, which is 1622(frame number)×2048(weight)×1200(height). We select one image in every 100 images as the test dataset. Chose one of them to train a model where K=8. The CR result is shown in Figure 9. The image id is after down sample by a factor 100 in order to get the test sample.

4.2 Analysis

For different models, LSTM is the closest to GRU, and the FC result is higher. LSTM and GRU are both gated neural network. GRU is a variant of LSTM, which has similar learning capabilities to LSTM in many tasks. Compared with FC, when dealing with data with sequence correlation properties, LSTM and GRU are more likely to find the law of the data and get better fitting results. The data trends in the figure are the same, indicating that the generalization capabilities of these three models are relatively similar. The compression rate fluctuates around 46%, and the highest is 52%. Compared to the common lossless image compression methods in figure 2, we get a reduction of more than 20% in data size.

5 Conclusion

We propose a lossless image compression method based on the neural network for SRS images. The result shows that our method can save more than 20% of storage space than common lossless image compression methods.
In the future, we will test more different network structures for lossless compression. Using video interpolation and video super-resolution methods to replace the difference module. Moreover, we will further explore the potential of image compression from space and time dimensions.

Reference

[1] Qi Fazhi, Huang Qiulan, Hu Hao, Tian Haolai, Wang Lu, Wang Yanming, Zhao Haifeng, Zhang Hongmei, Zeng Shan. The Design of Science Data Platform for High Energy Photon Source[J]. Frontiers of Data and Computing, 2020, 2(2): 40-58.

[2] W3C PNG Development Group (2003, Oct.). Portable Network Graphics (PNG) Specification (Second Edition), Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification. ISO/IEC 15948: 2003 (E). Available: http://www.w3.org/TR/PNG/

[3] Christopoulos C, Skodras A S, Ebrahimi T. The JPEG2000 still image coding system: An overview[J]. IEEE Transactions on Consumer Electronics, 2000, 46(4):1103-1127.

[4] Sneyers J, Wuille P. FLIF: Free lossless image format based on MANIAC compression[C]// 2016 IEEE International Conference on Image Processing (ICIP). IEEE, 2016.

[5] Hori T, Cho J, Watanabe S. End-to-end Speech Recognition with Word-based RNN Language Models[J]. 2018.

[6] Wael Farhan, Bashar Talafha, Analle Abuammar, et al. Unsupervised dialectal neural machine translation[J]. Information Processing and Management,2020,57(3).

[7] Goyal M, Tatwawadi K, Chandak S, et al. DeepZip: Lossless Data Compression using Recurrent Neural Networks[J]. 2018.