Supplementary Information for

"PC-bzip2: a phase-space continuity enhanced lossless compression algorithm for light field microscopy data"

Supplementary Information Table

Supplementary Figure 1	Comparison of pc-bzip2 decompressed image and pc-bzip2 compression input image
Supplementary Figure 2	Pipeline of the pc-bzip2 lossless compression algorithm
Supplementary Figure 3	Pipeline of the pc-bzip2 extended to time dimension compression algorithm
Supplementary Figure 4	The functions used for LFM data compression with the temporal continuity introduced.
Supplementary Note 1	LFM optical setup
Supplementary Note 2	Samples preparation
Supplementary Note 3	Extension to time series data
Supplementary Note 4	Performance comparison of different compression methods
Supplementary Note 5	Optimization for varying computational resources
Supplementary Table 1	Optical objective and illumination source usages in experiments.

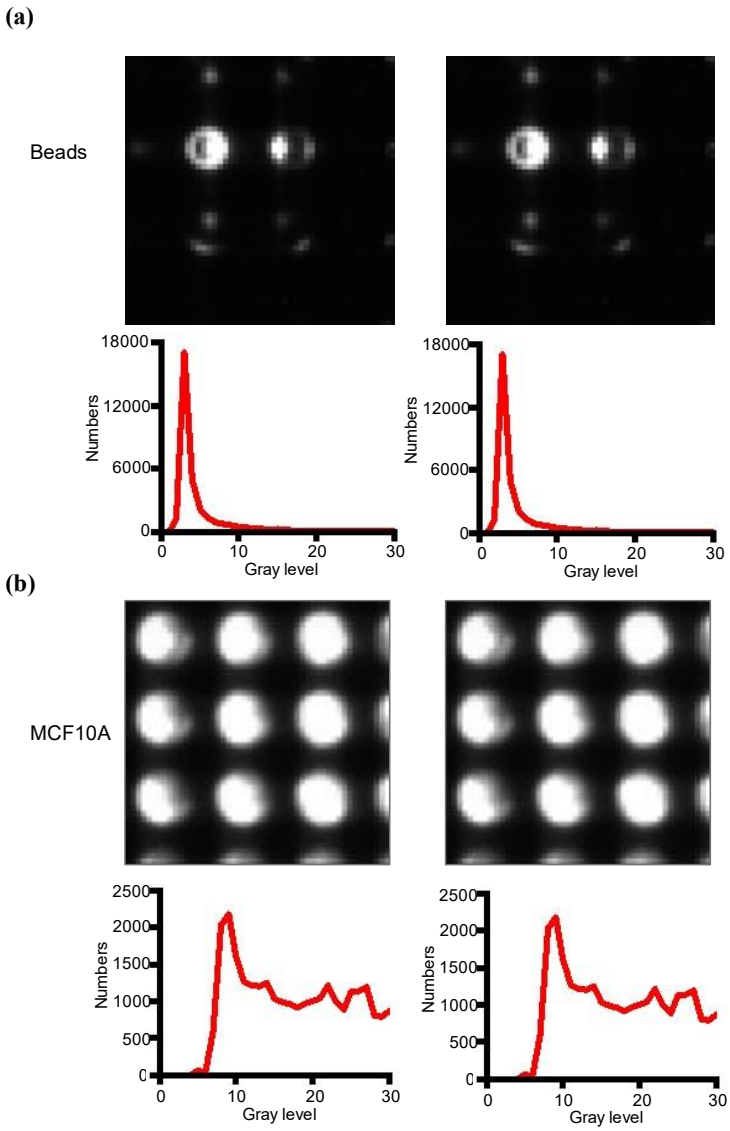


Figure S1. Comparison of pc-bzip2 decompressed image and pc-bzip2 compression input image. The left column is the pc-bzip2 compression input images and its grayscale histogram. The right column is the pc-bzip2 decompressed image and its grayscale histogram. (a) A part of fluorescent bead image with exposure times of 1024 ms. (b)A part of MCF-10A cells image with exposure times of 1024 ms.

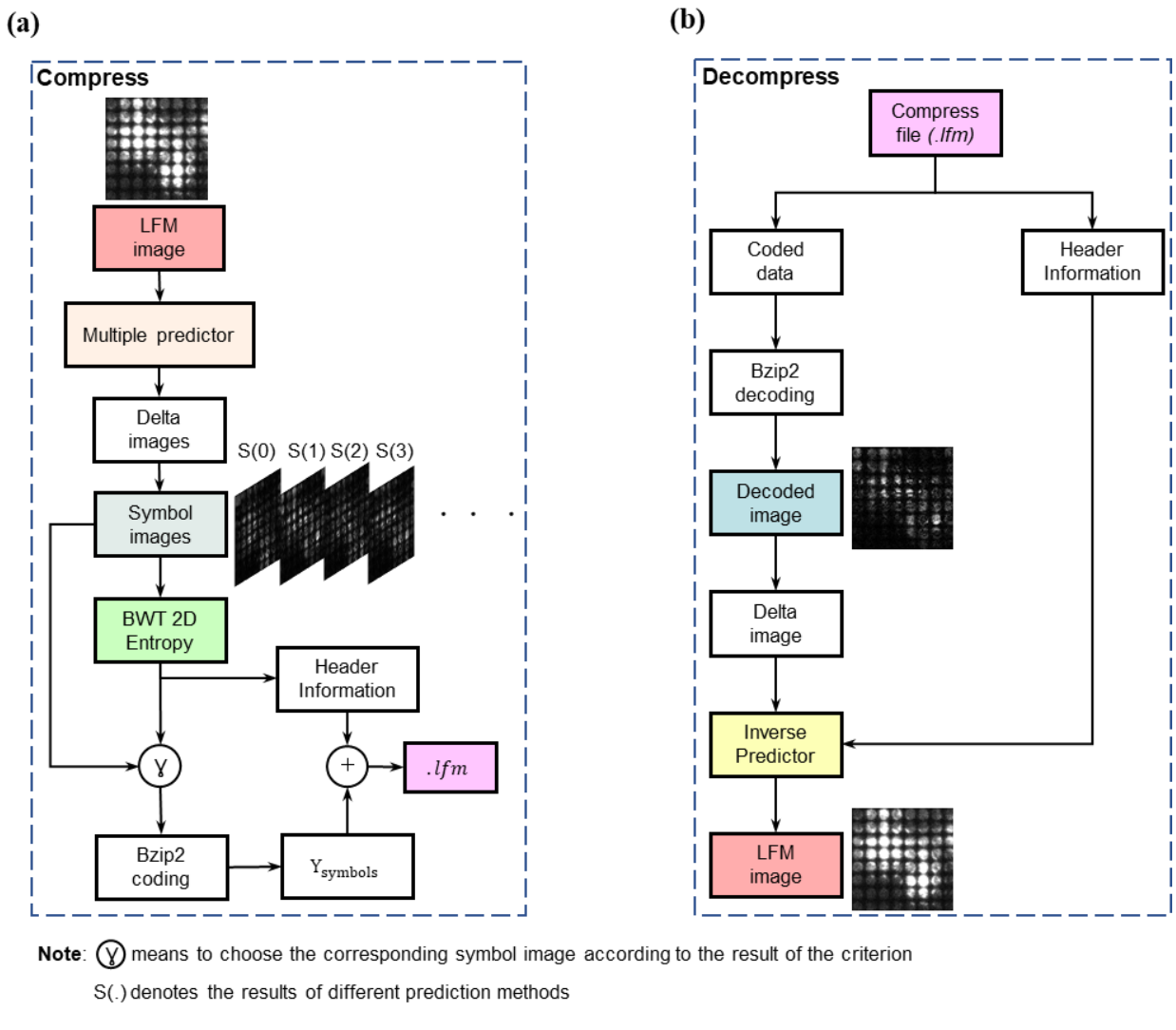


Figure S2. Pipeline of the pc-bzip2 lossless compression algorithm. (a) The pipeline of the pc-bzip2 lossless compression with typical examples of the output for the main step. Multiple predictors step means to perform all prediction methods individually and symbol images step means to convert the delta images to full positive images. **(b)** The pipeline of the pc-bzip2 decompression with typical examples of the output for the main step. We need to decide which inverse prediction method to use based on the header information. The decoded image here is one of the symbol images in compression pipeline.

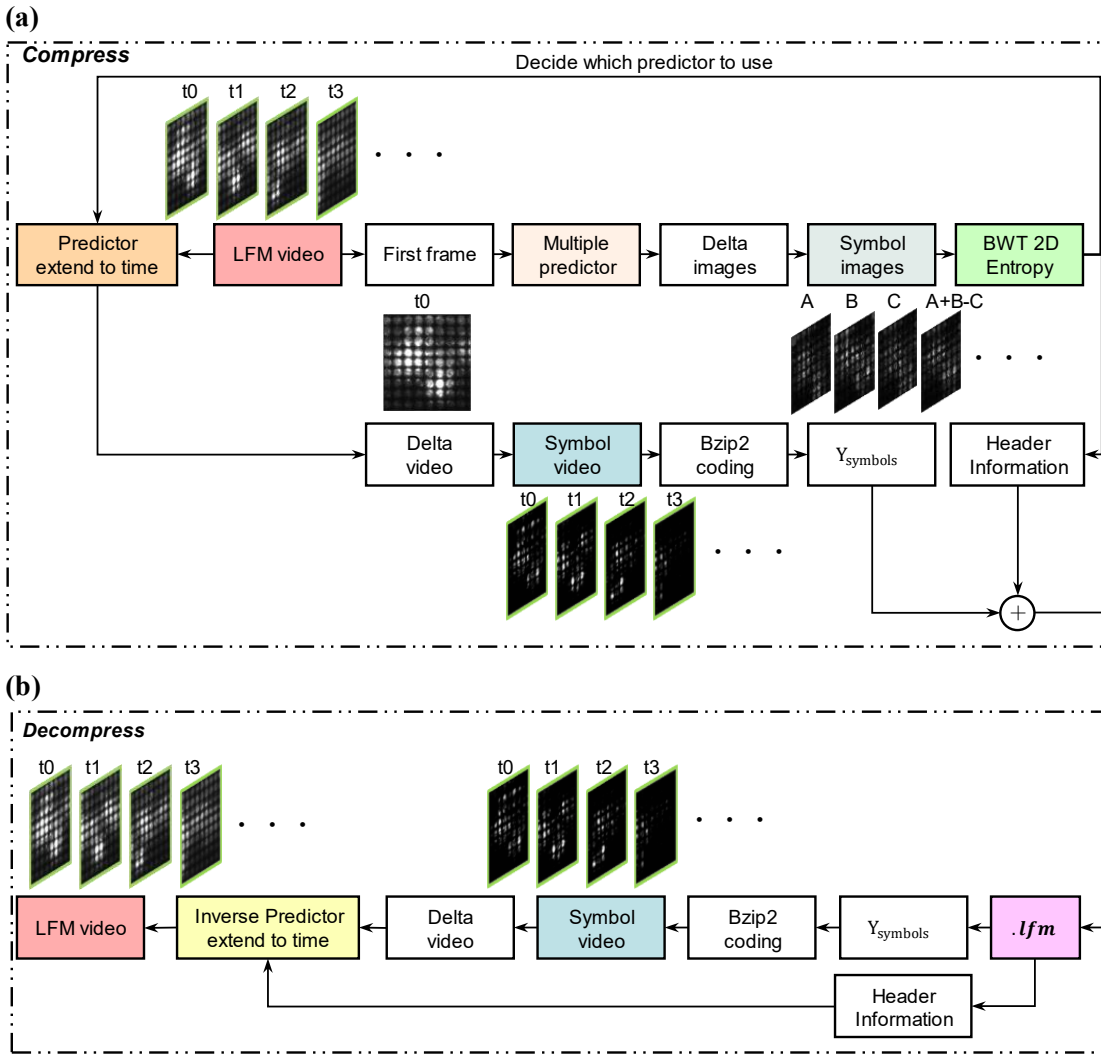
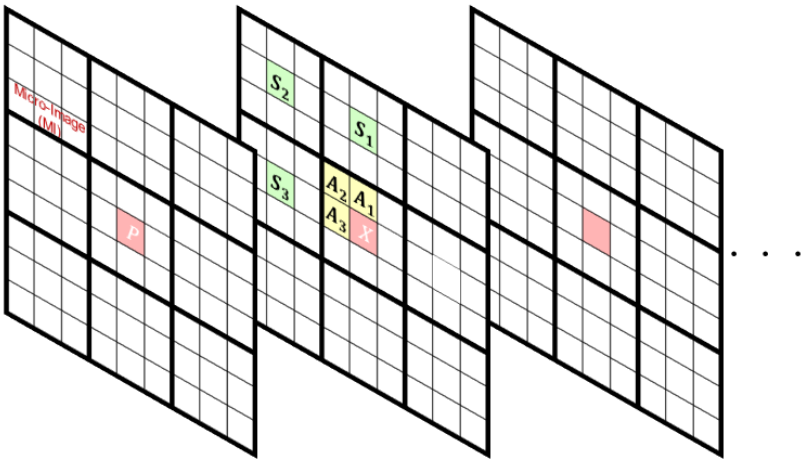


Figure S3. Pipeline of the pc-bzip2 extended to time dimension compression algorithm. (a) The pipeline of the pc-bzip2 extended to the time dimension compression with typical examples of the output for the main step. The first frame of video uses the pc-bzip2 compression to find the best predictor according to the two dimensions image entropy criterion, and then the subsequent frames will be predicted based on this predictor combined with time continuity. **(b)** The pipeline of the pc-bzip2 extended to the time dimension decompression with typical examples of the output for the main step. Which inverse predictor to be used depends on the header information.

(a)



(b)

predictor	function
A+P	$X = \left(\left((S_3 + A_3) \gg 1\right) + \mathrm{P}\right) \gg 1$
B+P	$X = \left(\left((S_1 + A_1) \gg 1\right) + \mathrm{P}\right) \gg 1$
C+P	$X = \left(\left((S_2 + A_1) \gg 1\right) + \mathrm{P}\right) \gg 1$
A+B-C+P	$X = \left(\left(\left(f_1(\mathrm{A}_1, A_2, \mathrm{A}_3) + f_1(S_1, S_2, S_3)\right) \gg 1\right) + P\right) \gg 1$
A+(B-C)/2+P	$X = \left(\left(\left(f_2(\mathrm{A}_1, A_2, \mathrm{A}_3) + f_2(S_1, S_2, S_3)\right) \gg 1\right) + P\right) \gg 1$
B+(A-C)/2+P	$X = \left(\left(\left(f_3(\mathrm{A}_1, A_2, \mathrm{A}_3) + f_3(S_1, S_2, S_3)\right) \gg 1\right) + P\right) \gg 1$
(A+B)/2+P	$X = \left(\left(\left(f_4(\mathrm{A}_1, A_2, \mathrm{A}_3) + f_4(S_1, S_2, S_3)\right) \gg 1\right) + P\right) \gg 1$

Figure S4. The functions used for LFM data compression with the temporal continuity introduced. (a) Schematic of time series LFM data structure. S_1, S_2, S_3, and X are adjacent pixels in spatial domain, A_1, A_2, A_3, and X are adjacent pixels in angular domain, P and X are adjacent pixels in temporal domain. **(b)** The functions of predictors used in phase-space continuity are combined with time continuity, where the f_1, f_2, f_3, f_4 here are the same as the functions in **Figure 4(b).**

Supplementary Table 1

Optical objective and illumination density in experiments

Figure	**Objective**	**Input illumination density**
Figure 6(a) & Figure 6(c) Zebrafish with high power	Zeiss 20x, NA 0.5 objective	Mercury lamp, 16.1 mw mm^{-2}
Figure 4(a) & Figure S1	Zeiss 63×, NA 1.25 objective	MCF10A(mCherry) 1.2 mw mm^{-2} Beads(GFP) 0.52 mw mm^{-2}
Figure 6(d) Zebrafish with low power	Zeiss 20x, NA 0.5 objective	Mercury lamp, 3.2 mw mm^{-2}
Figure 5(a)	Zeiss 20x, NA 0.5 objective	Mercury lamp, 16.1 mw mm^{-2} & 3.2 mw mm^{-2}
Figure S5(a) & Figure S5(b)	Zeiss 40x, NA 0.75 objective	Mercury lamp, 2.3 mw mm^{-2}

Supplementary Note 1: LFM optical setup

We develop the light field microscope based on an epi-fluorescence microscope (Zeiss, Axio Observer 7), which includes a mercury lamp and BFP, GFP and mCherry filter cube sets. We use a 20× 0.5NA air-immersion objective lens (Zeiss Objective EC Plan-Neofluar 20×/0.50 M27) and a 63×/1.25NA oil-immersion objective lens (Zeiss Objective EC PN 63×/1.25 Oil M27) in our experiments. Besides, the light field microscope also contains an optical collection arm, which is aligned with the BX43 light outlet. In the optical collection arm, a microlens array (RPC Photonics, MLA-S100-f21) is placed at the image plane of BX43. The focal plane of microlens array is conjugated to the imaging sensor plane by using a 1:1 relay lens pair. The imaging sensor in use is an sCMOS camera with 2048 × 2048 pixels (PCO Panda4.2) (**Fig 1a**).

Supplementary Note 2: Samples preparation

Fluorescence beads imaging

Green fluorescent beads with 2 um diameter (1:1000 diluted in PBS, ThermoFisher) were embedded in 1% low-melting-temperature agarose, dropped on the glass slide surface, and covered with 0.17 mm cover glass. We imaged the beads with the self-built light field microscope at room temperature (RT).

MCF10A cell fluorescent staining and imaging

MCF-10A cells were obtained from ATCC (CRL-10317) and preserved in complete BMEM medium (Lonza, CC-3150). To image these samples, 2×10^4 cells were seeded on Φ12 mm circular coverslips (Marienfeld 0117520), which was pre-treated 1 h with 0.1% PDL (Sigma P6407) at RT. The cells were returned into CO2 cell incubator overnight. Cells were then washed with PBS (GIBCO), fixed in 4% PFA (Solarbio) for 15 min, permeated and blocked with 0.2% TritonX-100 (Sigma T8787) in 3% BSA (Amresco 0332) PBS buffer at RT for 30 min. After that, cells were incubated with 1:1000 diluted TOM20 antibody (Santa Cruz sc-11451) at 4◦C. The next day, cells were washed with 0.05% PBST (0.05% Tween-20 (Sigma P7949) in PBS), and stained with 1:200 diluted Alexafluo555-labeled secondary antibody (CST 4413S) for 2 h, accompanied with 1:200 diluted Alexafluo488-labeled phalloidin (CST 8878S) at RT. After being thoroughly washed with PBS, cells were mounted by Mowiol mounting medium containing 2μg/mL DAPI, and stored away from light at 4◦C. The cells were kept at room temperature and imaged at different exposure time.

Zebrafish blood cell imaging

Zebrafish belonging to the transgenic line Tg(flk:EGFP) were employed for imaging blood cells, The resulting embryos were reared at 28.5°C until 4 days post-fertilization (dpf). To immobilize the larval zebrafish, they were briefly immersed in a $1 mg ml^{-1}$ α-bungarotoxin solution (Invitrogen). The paralyzed larvae were then embedded in 1% low-melting-temperature agarose inside a Glass Bottom Dish (35 mm Dish with 20 mm Bottom Well, Cellvis). The specimens were kept at room temperature and the zebrafish larvae were imaged at 40 Hz.

Supplementary Note 3: Extension to time series data

As shown in **Fig. S4**, for the corresponding pixel X, it not only has angle-adjacent pixels S_1, S_2, S_3, and spatially adjacent pixels A_1, A_2, A_3, but also has temporally adjacent pixel P. Therefore, the value of pixel X can be predicted by considering both of these continuities. This process can be simply expressed as:

$$A(X) = F_i(S_1, S_2, S_3, A_1, A_2, A_3, P) \quad (1)$$

where $A(X)$ is the predicted value, F_i represents the i-th predictor with the temporal continuity introduced (Predictor $A + P$ corresponds to the 1-th predictor, while predictor $(A + B)/2 + P$ corresponds to the 7th predictor) as shown in **Fig. S4b**. In order to find the best predictor for a single frame, the first step is to calculate the two-dimensional entropy for the first frame image to obtain its corresponding optimal prediction method. This can be mathematically described as:

$$d = 2D\_entropy(I_0) \quad (2)$$

where I_0 is the first frame image, $2D\_entropy()$ is the two-dimensional image entropy criterion corresponding to the Eq. (2) to Eq. (6) in the main text, and d is the index of the optimal predictor. The result d of the two-dimensional entropy criterion will be used as the main predictor in the subsequent frames (For example, if $d = 7$, then the subsequent predictor is $(A + B)/2 + P$). the subsequent frames, with the index $i = d$, the predictor can be simply expressed as:

$$A(X) = F_d(S_1, S_2, S_3, A_1, A_2, A_3, P) \quad (3)$$

Then, adopting a compression strategy akin to image compression, we initially derive the differential data of the video and proceed with compression using the bzip2 algorithm.

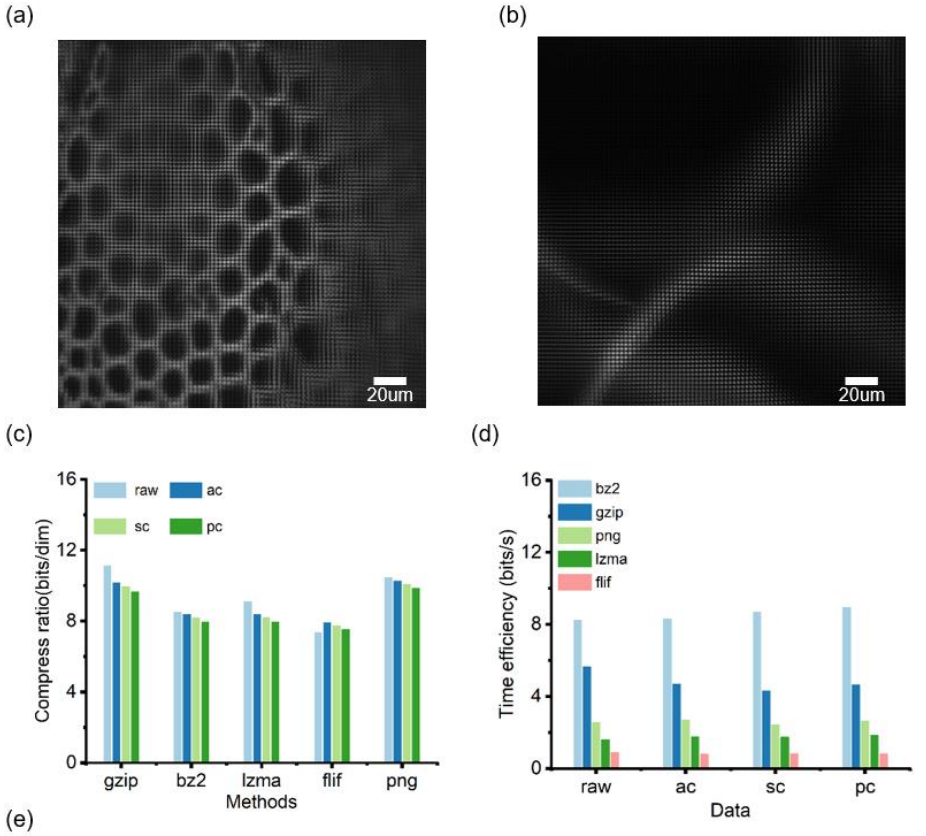
Supplementary Note 4: Performance comparison of different compression methods

We investigated commonly used lossless compression methods that support 16-bit images, mainly including traditional methods such as gzip, bzip2, lzma, flif, and png, Additionally, we also surveyed learning-based methods such as L3C[39], RC[40], Bit-Swap[29], HiLLoC[27], IDF[35], IDF++[41], LBB[30], IVPF[42] and iFlow[43]. However, these learning-based methods are often designed for 8-bit RGB images (3 channel, totally 24-bit) and do not offer support for 16-bit images. Therefore, our main focus will be on comparing bzip2 with commonly used non-learning-based methods. The datasets used in this experiment comprise different MCF10A data captured under different exposures (including 64ms (M_64), 128ms (M_128), 256ms (M_256), 512ms (M_512), and 1024ms (M_1024), as shown in figure 4(a)), as well as collected plant sample data (cotton stem (CS), as shown in Figure S5(a), and dandelion fluff (DF), as shown in Figure S5(b)). Taking the data of MCF10A cells with 64ms exposure time as an example, we compared the compression performance of different compression methods (including gzip, bzip2, png, lzma, and flif) under various prediction modes. On the M_64 data, we compared the compression ratio under different prediction modes, demonstrating that the introduction of phase space continuity can indeed further reduce information redundancy compared to using spatial or angular continuity alone as shown in Figure S5(c). Comparisons of different compression methods also showed that bzip2 exhibits superior compression ratio, as shown in Figure S5(c). To further compare the time efficiency of different compression methods, we defined time efficiency as:

$$T_e = \left(b_{raw} - b_{comp}\right)/t_{comp} \tag{4}$$

where b_{raw} is the number of bits required per pixel in the original image, b_{comp} is the number of bits required per pixel after compression, and t_{comp} is the time consumption for compression. Therefore, T_e can represent the encoding efficiency per unit of time. The comparison of different compression methods demonstrated that bzip2 exhibits exceptional time efficiency, as shown in Figure S5(d). On the contrary, FLIF achieves improved compression ratios at the expense of additional time consumption. To further validate the role of phase-space continuity, we conducted a comprehensive statistical analysis of all data to compare the improvement in compression ratios achieved by utilizing phase-space continuity against not utilizing it, as illustrated in Figure S5(e). The results indicate that for encoding methods, the introduction of phase-space continuity significantly enhances compression ratios. However, for universal image formats

like PNG and FLIF, which inherently incorporate predictive operations, external additional operations lead to increased data dispersion, resulting in a decrease in compression ratios.



(e)

data	gzip	bz2	lama	flif	png
M_64	1.45	0.54	1.17	-0.19	0.61
M_128	1.53	0.81	1.33	-0.17	0.69
M_256	1.48	0.99	1.37	-0.17	-0.29
M_512	1.46	1.18	1.55	-0.12	0.60
M_1024	1.46	1.31	1.73	-0.13	-0.18
DF	1.39	0.80	1.31	-0.05	-0.26
CS	1.55	1.72	2.05	-0.18	0.25

Figure S5. Performance comparison of different compression methods. **(a)** The light-filed image of the cotton stem (CS). **(b)** The light-filed image of Dandelion fluff (DF). **(c)** Comparison of compression ratios of different compression methods under different prediction modes. **(d)** Comparison of time efficiency of different compression methods under different prediction modes. "raw" means no prediction, "sc" means using space prediction, "ac" means using angle prediction, "pc" means using phase-space prediction. **(e)** The improvement in compression ratio (bits/dim), when utilizing phase space continuity across different data for various compression methods compared to those without.

Supplementary Note 5: Optimization for varying computational resources

PC-Bzip2 aims to reduce time consumption mainly by fully utilizing the system's parallel processing capabilities, including multi-threaded processing by the CPU and parallel acceleration by the GPU. The GPU acceleration performance in PC-Bzip2 typically depends on the computational capabilities of the GPU hardware itself. For different GPUs, optimizations can be attempted to further reduce time consumption by optimizing data transfers between CPU and the GPU, optimizing thread block allocation, and introducing the use of shared memory, et al.

Regarding the CPU acceleration performance in PC-Bzip2, it is mainly influenced by three factors: (1) the maximum number of threads supported by the CPU; (2) the size of data processed by each thread; (3) the limitation by the hardware system's memory size. PC-Bzip2 defaults to using the maximum number of threads supported by the CPU. However, when the memory size of the hardware system is limited, adjustments to the number of CPU threads must be made to ensure that the overall memory consumption remains within the system's supported range. Additionally, the size of data processed by each thread needs to be appropriately configured based on the CPU's computational capabilities. Setting it excessively large or small can both result in increased time consumption.