Supplementary Information for

Carbon nanotube analog tensor core accelerating edge computer vision

Authors

Jingfang Pei[1,#], Lekai Song[1,2,#], Songwei Liu[1,#], Yingyi Wen[1], Yang Liu[1,3], Pengyu Liu[1], Wenyu Cui[2], Xin Lu[5], Teng Ma[2], Zegao Wang[5], Guohua Hu[1,*]

Affiliations

[1]Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong S. A. R., 999077, China

[2]Department of Applied Physics, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong S. A. R., 999077, China

[3] Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong S. A. R., China

[5]College of Materials Science and Engineering, Sichuan University, Chengdu, 610065, China

# These authors contribute equally

*Correspondence email: ghhu@ee.cuhk.edu.hk

This file contains:

Supplementary Note 1

Supplementary Figure S1-S22

Supplementary Table S1-S2

Supplementary References

Supplementary Note 1: Energy consumption estimation

When operating our NVMs to process the data, the energy consumption arises from two main processes: i) *Programming*: gate pulses V_{gs} are applied to program the memory weights; ii) *Data processing*: analog source-drain pulses V_{ds} are applied to perform multiply-accumulation processing of the data.[1] We now discuss the energy consumption of these two processes:

- ***Programming***

The energy consumption of one single memory weight programming can be estimated by

$$E_{\mathrm{programming}} = V_{\mathrm{gs}} \times I_{\mathrm{gs}} \times t_{\mathrm{pulse}}.$$

Using our multi-state pulsed memory weight programming method (Fig. 2i and Fig. S5), the duration of each pulse is set as 500 ns, the maximum operating voltage V_{gs} is 7.5 V, and the corresponding gate current I_{gs} is 1.06×10^{-9} A (read from the transfer characteristics). Therefore, we estimate the energy consumption for one single weight programming as

$$E_{\mathrm{max,programming}} = 7.5 \times 1.06 \times 10^{-9} \times 500 \times 10^{-9} = 3.98 \times 10^{-15}\ \mathrm{J} = 3.98\ \mathrm{fJ}.$$

- ***Data processing***

During data processing, no V_{gs} pulses are applied after the programming, and analog V_{ds} pulses are applied. To process one single bit of data, the energy consumption can be estimated by

$$E_{\mathrm{data\ processing}} = V_{\mathrm{ds}} \times I_{\mathrm{ds}} \times t_{\mathrm{pulse}}.$$

In our demonstration (Fig. S6-7), the data processing is at a frequency of 1 MHz, the duration of one single bit is 1 μs, the on-state conductance of our NVMs is 30.8 μS (read from the memory weight programming, Fig. 2i), and the corresponding current is 3.08×10^{-5} A at the maximum input voltage of 1 V. Therefore, we estimate the energy consumption for processing one single bit as

$$E_{\mathrm{max,data\ processing}} = 1 \times 3.08 \times 10^{-5} \times 1 \times 10^{-6} = 3.08 \times 10^{-11}\ \mathrm{J} = 30.8\ \mathrm{pJ}.$$

Therefore, based on the above estimations, the energy consumption of our NVMs is ~3.98 fJ for memory weight programming and ~30.8 pJ for data processing.

Supplementary Figures

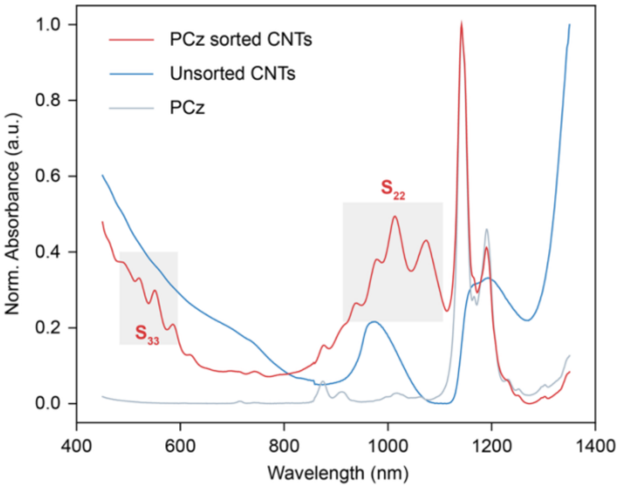


Figure S1. UV-Vis-NIR absorption spectra of the CNT and sorting polymer PCz solutions. The S33 and S22 peaks prove successful sorting of the semiconducting single-walled CNTs by the sorting polymer PCz.[2]

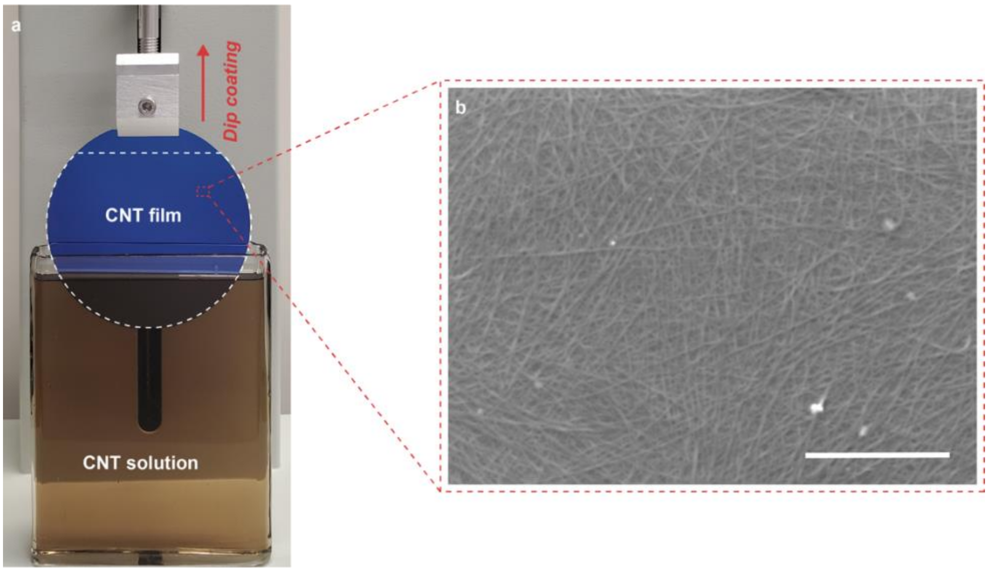


Figure S2. CNT thin film deposition. (a) CNT film deposition on 4-inch SiO_2/Si wafer by dip coating of the PCz sorted CNT solution. (b) Scanning electron microscopic (SEM) image of the CNT film on SiO_2/Si wafer, showing uniform CNT film with high density. Scale bar – 1 µm.

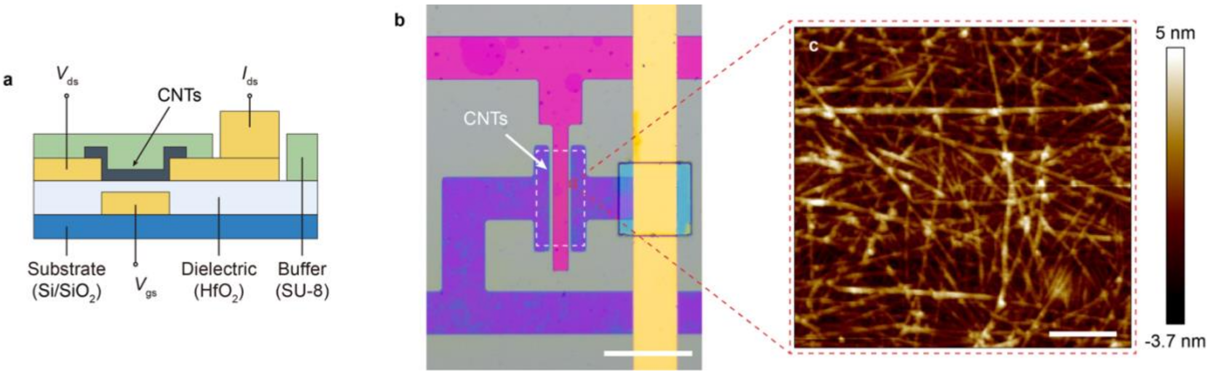


Figure S3. Microscopic characterization of the CNT NVMs. (a) Schematic structure of the CNT NVM devices. (b) Optical microscopic image of a typical NVM device in crossbar array, and (c) atomic force microscopic (AFM) image of the CNT channel. Scale bars – (b) 100 µm, (c) 200 nm.

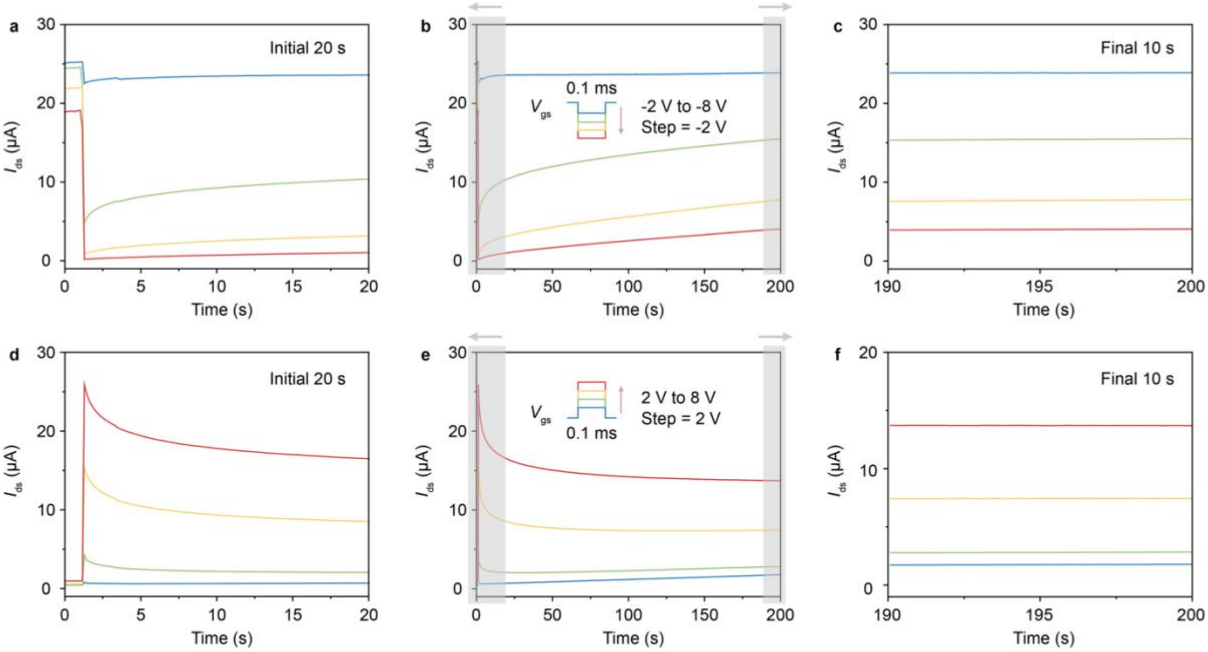


Figure S4. Long-term memory retention of the CNT NVMs under pulsed gate voltage signals. Long-term memory retention of a typical NVM device under one single gate pulse V_{gs} in (a-c) negative amplitudes and (d-f) positive amplitudes. The gate pulse V_{gs} settings are shown in (b) and (e). The drain-source pulses $V_{\mathrm{d}s}$ are set as 0.5 V DC in all the tests. (a, d) The retention behavior during the initial 0 to 20 s. (c, f) The stabilized retention behavior during the final 190 to 200 s. The tests prove a long-term memory retention of our CNT NVMs.

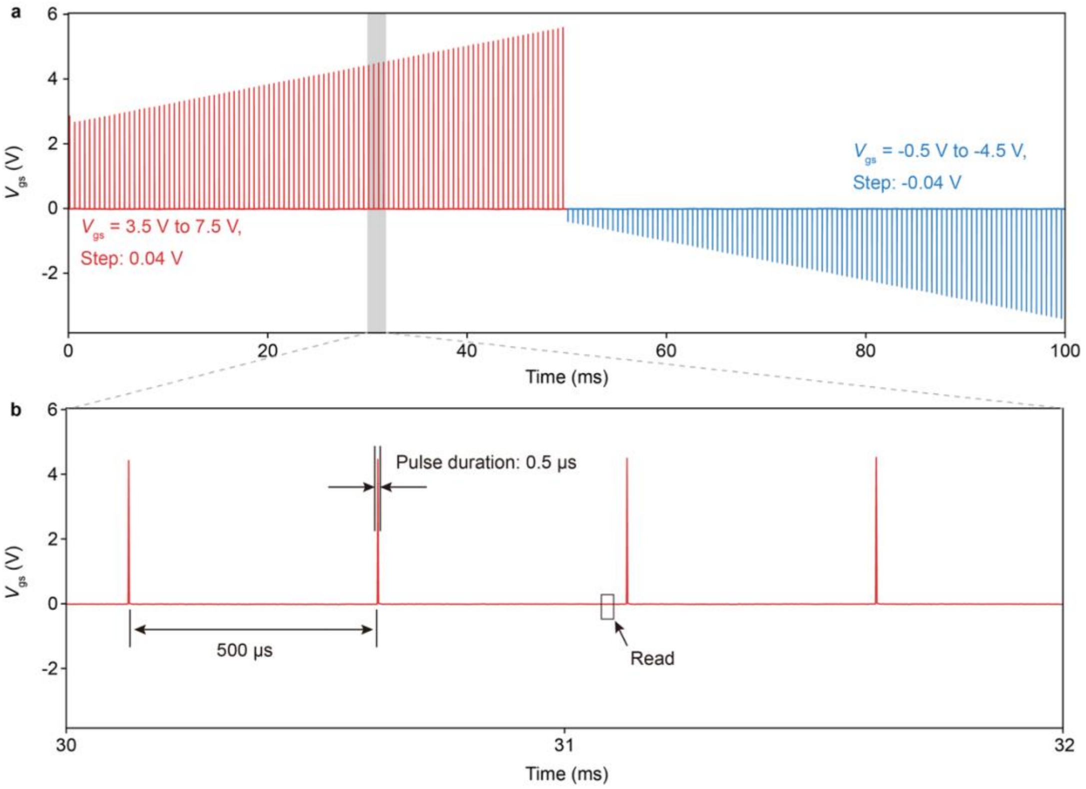


Figure S5. Pulse train of gate voltage signal V_{gs} to program the memory weight of the CNT NVMs. (a) Pulse train V_{gs}, and (b) the detailed pulsed signal parameter and the memory weight read operation. The drain-source pulses V_{ds} are set as 0.5 V DC in the tests. The pulse train V_{gs} enables linearity and symmetry in the potentiation and depression operations of our CNT NVMs.

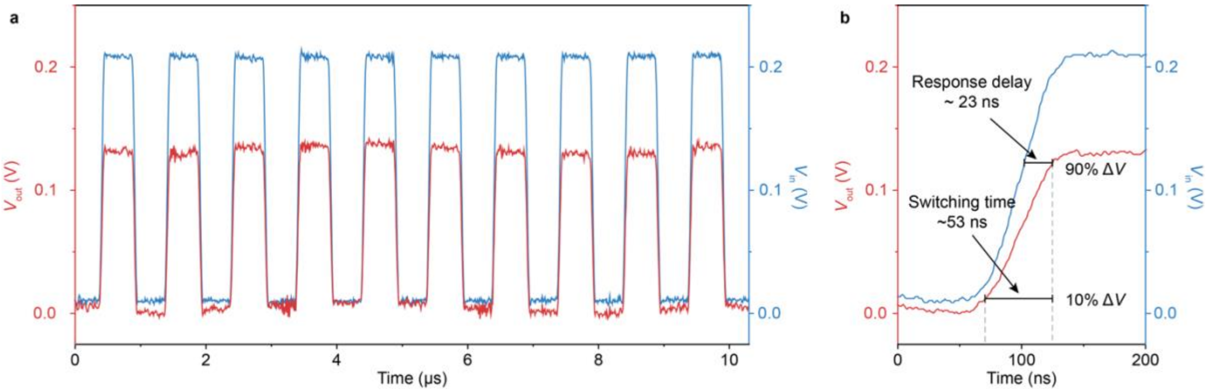


Figure S6. Response of the CNT NVMs. (a) 10-cycle response of a typical NVM device under drain-source pulses V_{ds} of 0.2 V amplitude at a frequency of 1 MHz, and (b) the detailed rising edge of the output, showing the response delay and switching time are ~23 ns and ~53 ns, respectively, proving a high data processing speed of our CNT NVMs. During the test, the NVM is programmed at the initial memory weight state.

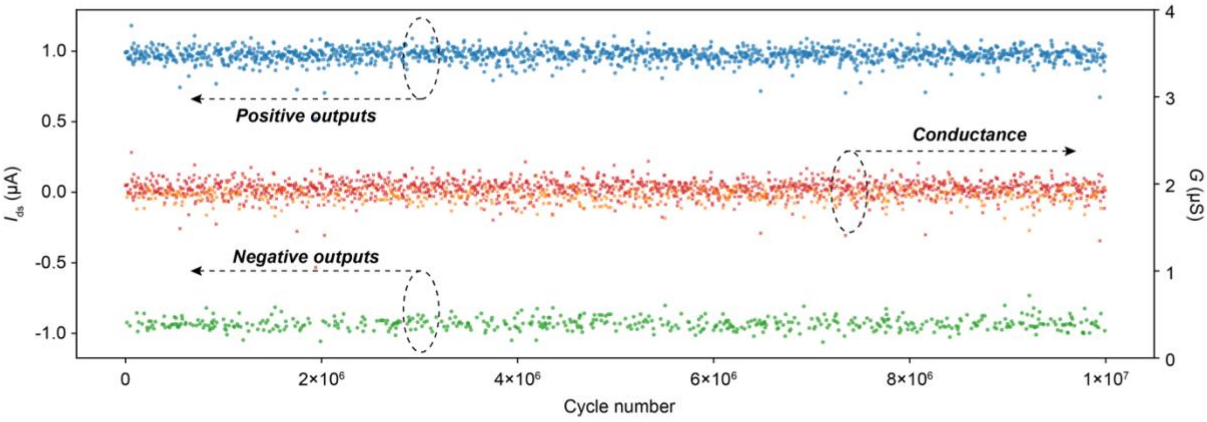


Figure S7. Endurance test of the CNT NVMs. A train of alternatively switching paired pulsed source-drain voltage signal V_{ds} of +/- 0.5 V is applied to the drain terminal. The pulsed signal frequency is 1MHz, corresponding to a data processing speed of 1 Mbit/s. During the test, the device is programmed at the initial memory weight state. The device undergoes 10^7 consecutive pulsed cycles. The current outputs I_{ds} in positive and negative amplitudes to the paired pulsed V_{ds} signals in each of the cycles are shown as the blue and green dots, respectively, and the corresponding conductance are shown as the red and orange dots, respectively. The stable memory (i.e. the conductance) behavior under the alternatively switching paired positive and negative pulsed V_{ds} signals proves that our CNT NVMs can undergo stable data processing without shifting the memory weight.

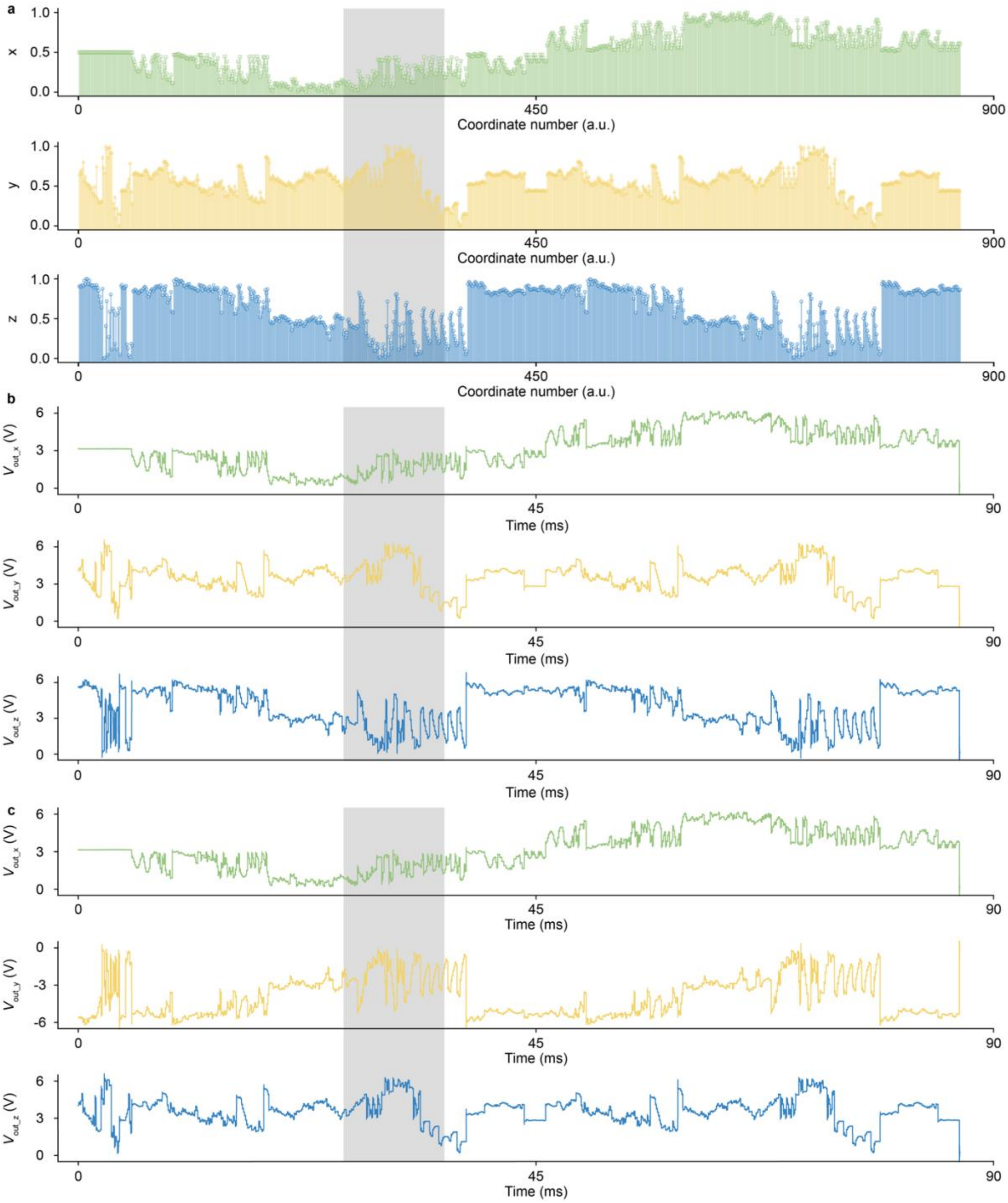


Figure S8. Experimental inputs and outputs of spatial transformation of the 3D model shown in Fig. 3c. (a) x-, y-, and z- input signals of a total of 837 coordinates of the 3D model. (b) Output signals to the coordinates after the identical spatial transformation operation (i.e. 0^{o} rotation around the x- y- and z-axes). The nearly identical output waveforms with input in (a) proves successful spatial data processing. (c) Output signals to the coordinates after 90^{o} rotation operation around the x-axis. The amplitudes in (a) correspond to the coordinate values of the model. The amplitudes in (b) and (c) are output voltages corresponding to the coordinate values after spatial transformation. The output voltages are sampled according to the input frequency intervals, and are then normalized and reconstructed to output the coordinate values. The gray regions in (a), (b) and (c) correspond to the regions presented in Fig. 3c (1-3), respectively.

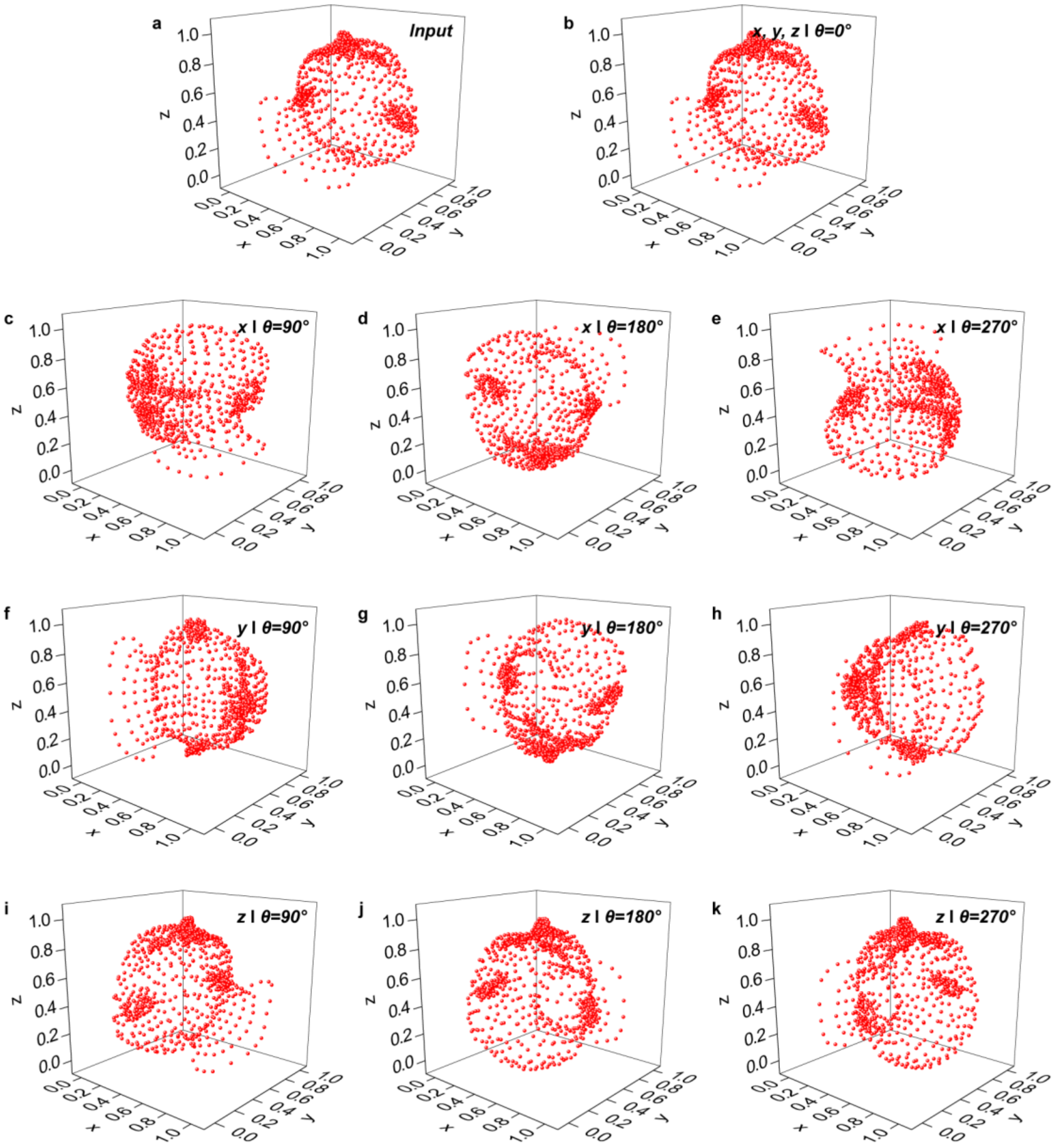


Figure S9. Experimental data points plotted in space coordinate system of the 3D model shown in Fig. 3d. (a) The input coordinates and (b) output coordinates after the identical spatial transformation operation. The outputs with 90°, 180° and 270° rotation operations around (c-e) x-, (f-h) y-, and (i-k) z- axes, respectively. See Movie S1 for the spatial transformation.

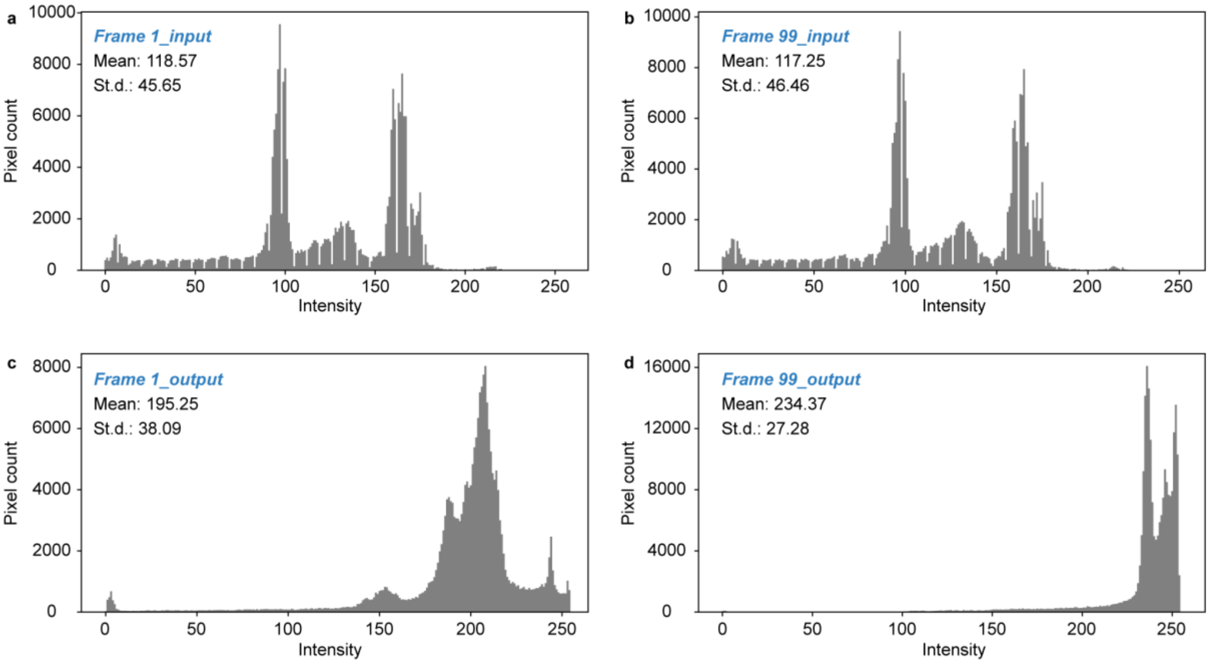
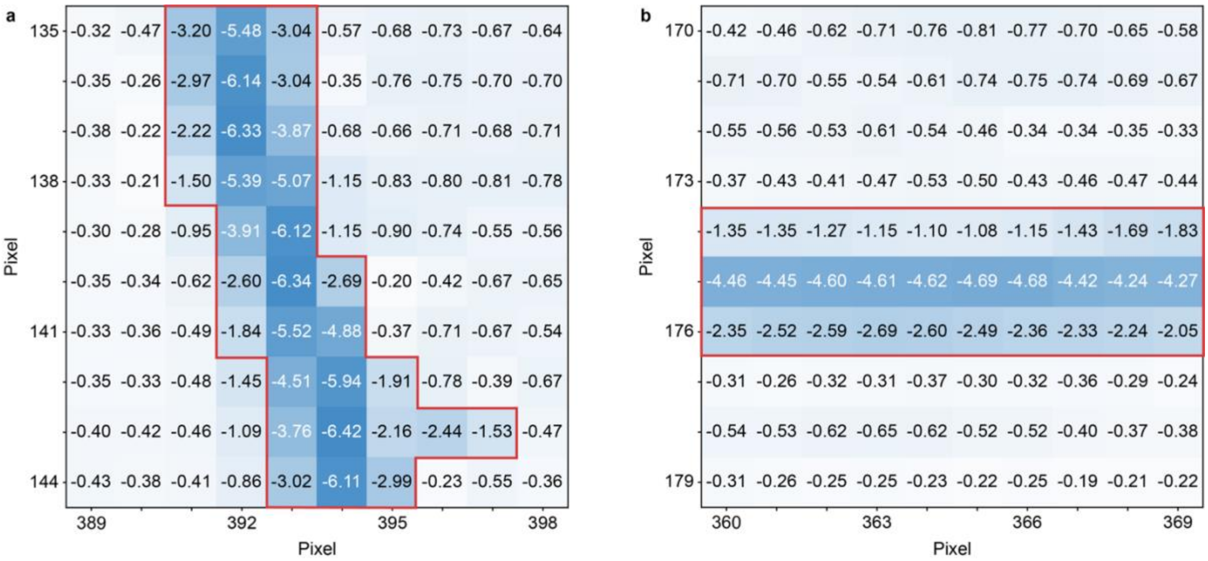


Figure S10. Distribution of pixel intensity in Fig. 4b. The distribution of pixel intensity in (a) input frame 1 and (b) frame 99, and in the edge detected (c) output frame and (d) frame 99. The distributions of the input frames in (a) and (b) show broader intensity spectra, with the intensities distributed relatively evenly. After edge detection, most non-edge areas are assigned a value of "0", while a small number of pixels at the edges are assigned "1", as shown in Fig. 4a. This process compresses the distribution spectra and pushes for stronger pixel intensities, as shown in (c) and (d). In addition, as demonstrated, the standard deviation of the pixel intensity decreases significantly, proving the effectiveness of the edge detection using our tensor core. Note that, due to the design of our back-end circuits, the output readout voltages are negative and as such, the reconstructed images show inverted intensity: the edges appear dark and the non-edge regions bright, opposite to the typical algorithm output. This inversion does not affect the effectiveness of edge detection.



	389			392			395			398
135	-0.32	-0.47	-3.20	-5.48	-3.04	-0.57	-0.68	-0.73	-0.67	-0.64
	-0.35	-0.26	-2.97	-6.14	-3.04	-0.35	-0.76	-0.75	-0.70	-0.70
	-0.38	-0.22	-2.22	-6.33	-3.87	-0.68	-0.66	-0.71	-0.68	-0.71
138	-0.33	-0.21	-1.50	-5.39	-5.07	-1.15	-0.83	-0.80	-0.81	-0.78
	-0.30	-0.28	-0.95	-3.91	-6.12	-1.15	-0.90	-0.74	-0.55	-0.56
	-0.35	-0.34	-0.62	-2.60	-6.34	-2.69	-0.20	-0.42	-0.67	-0.65
141	-0.33	-0.36	-0.49	-1.84	-5.52	-4.88	-0.37	-0.71	-0.67	-0.54
	-0.35	-0.33	-0.48	-1.45	-4.51	-5.94	-1.91	-0.78	-0.39	-0.67
	-0.40	-0.42	-0.46	-1.09	-3.76	-6.42	-2.16	-2.44	-1.53	-0.47
144	-0.43	-0.38	-0.41	-0.86	-3.02	-6.11	-2.99	-0.23	-0.55	-0.36

	360			363			366			369
170	-0.42	-0.46	-0.62	-0.71	-0.76	-0.81	-0.77	-0.70	-0.65	-0.58
	-0.71	-0.70	-0.55	-0.54	-0.61	-0.74	-0.75	-0.74	-0.69	-0.67
	-0.55	-0.56	-0.53	-0.61	-0.54	-0.46	-0.34	-0.34	-0.35	-0.33
173	-0.37	-0.43	-0.41	-0.47	-0.53	-0.50	-0.43	-0.46	-0.47	-0.44
	-1.35	-1.35	-1.27	-1.15	-1.10	-1.08	-1.15	-1.43	-1.69	-1.83
	-4.46	-4.45	-4.60	-4.61	-4.62	-4.69	-4.68	-4.42	-4.24	-4.27
176	-2.35	-2.52	-2.59	-2.69	-2.60	-2.49	-2.36	-2.33	-2.24	-2.05
	-0.31	-0.26	-0.32	-0.31	-0.37	-0.30	-0.32	-0.36	-0.29	-0.24
	-0.54	-0.53	-0.62	-0.65	-0.62	-0.52	-0.52	-0.40	-0.37	-0.38
179	-0.31	-0.26	-0.25	-0.25	-0.23	-0.22	-0.25	-0.19	-0.21	-0.22

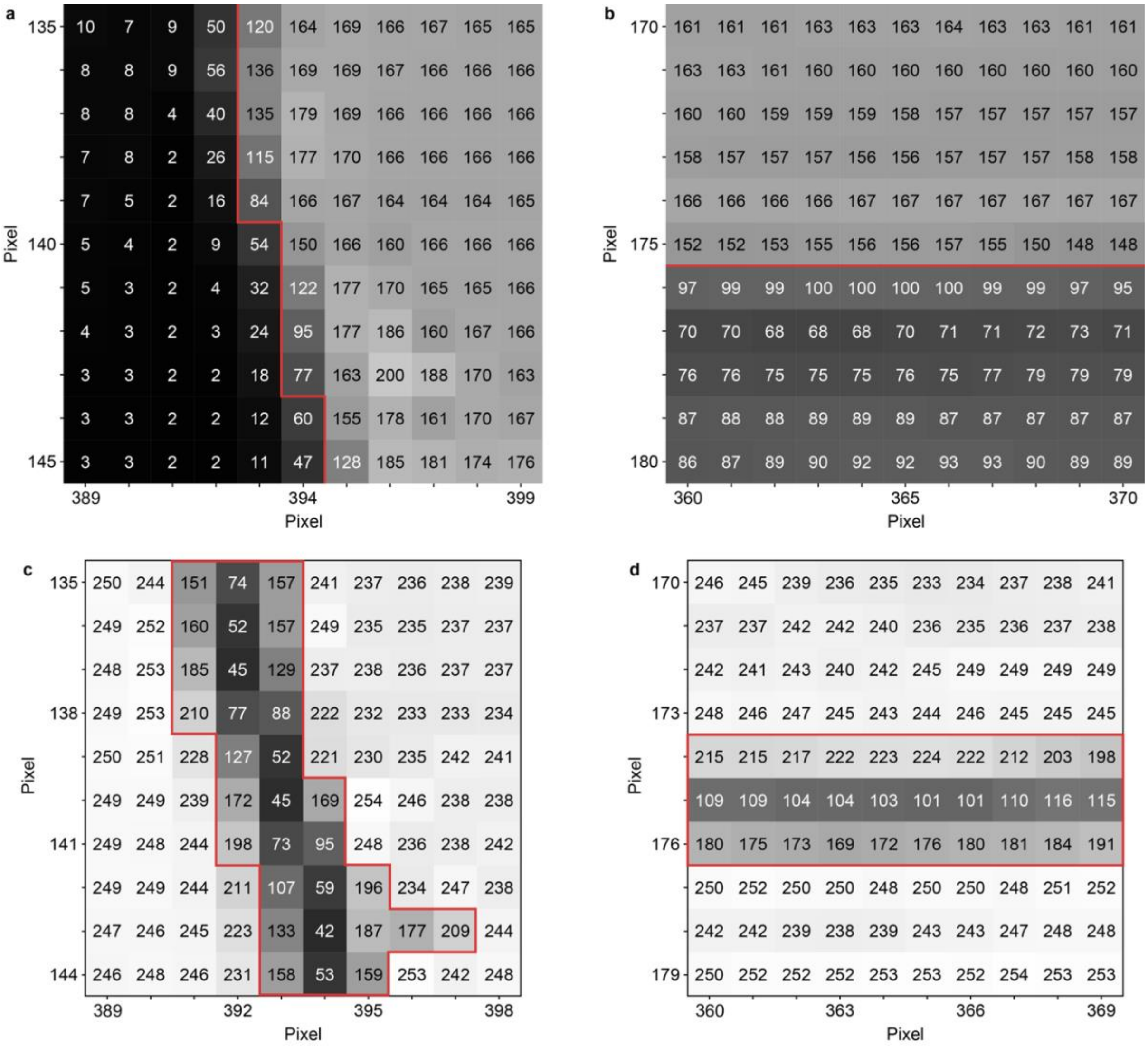
Figure S11. Experimental edge detection output heatmaps of frame 99 shown in Fig. 4c. An area of 10 × 10 pixels is used to evaluate the (a) vertical and (b) horizontal edge detection from 11 × 11 pixels input using our analog tensor core. The values presented in the heatmaps are the output voltages (V_{out}) from the tensor core, and the edges detected are enclosed by the red outlines. To evaluate the detection performance quantitatively, we estimate the signal-to-noise ratio (SNR) using

$$\text{SNR (dB)} = 20log_{10}(\frac{\mu_{\text{edge}}-\mu_{\text{background}}}{\sigma_{\text{background}}}),$$

where μ_{edge} is the averaged V_{out} in the edge region, while $\mu_{\text{background}}$ is the averaged V_{out} in the non-edge region, and $\sigma_{\text{background}}$ is the standard deviation (s. d.) of the V_{out} in the non-edge region. The SNR is estimated as 22.69 dB for region (a) and 22.64 dB for region (b), respectively. In addition, dynamic range is also used to evaluate the edge detection,

$$\text{Dynamic range (dB)} = 20log_{10}(\frac{V_{\text{max}}}{V_{\text{min}}}),$$

where V_{max} is the maximum V_{out} in the whole region, while V_{min} is the minimum V_{out}. The dynamic range is estimated as 30.13 dB for (a) and 27.85 dB for (b), respectively. The SNR and dynamic range, approaching the industrial application standard, prove the ability of our analog tensor core in performing efficient, low-pression edge detection scenarios at edge.



a

Pixel	389					394					399
135	10	7	9	50	120	164	169	166	167	165	165
	8	8	9	56	136	169	169	167	166	166	166
	8	8	4	40	135	179	169	166	166	166	166
	7	8	2	26	115	177	170	166	166	166	166
	7	5	2	16	84	166	167	164	164	164	165
140	5	4	2	9	54	150	166	160	166	166	166
	5	3	2	4	32	122	177	170	165	165	166
	4	3	2	3	24	95	177	186	160	167	166
	3	3	2	2	18	77	163	200	188	170	163
	3	3	2	2	12	60	155	178	161	170	167
145	3	3	2	2	11	47	128	185	181	174	176

b

Pixel	360					365					370
170	161	161	161	163	163	163	164	163	163	161	161
	163	163	161	160	160	160	160	160	160	160	160
	160	160	159	159	159	158	157	157	157	157	157
	158	157	157	157	156	156	157	157	157	158	158
	166	166	166	166	167	167	167	167	167	167	167
175	152	152	153	155	156	156	157	155	150	148	148
	97	99	99	100	100	100	100	99	99	97	95
	70	70	68	68	68	70	71	71	72	73	71
	76	76	75	75	75	76	75	77	79	79	79
	87	88	88	89	89	89	87	87	87	87	87
180	86	87	89	90	92	92	93	93	90	89	89

c

Pixel	389			392			395			398
135	250	244	151	74	157	241	237	236	238	239
	249	252	160	52	157	249	235	235	237	237
	248	253	185	45	129	237	238	236	237	237
138	249	253	210	77	88	222	232	233	233	234
	250	251	228	127	52	221	230	235	242	241
	249	249	239	172	45	169	254	246	238	238
141	249	248	244	198	73	95	248	236	238	242
	249	249	244	211	107	59	196	234	247	238
	247	246	245	223	133	42	187	177	209	244
144	246	248	246	231	158	53	159	253	242	248

d

Pixel	360			363			366			369
170	246	245	239	236	235	233	234	237	238	241
	237	237	242	242	240	236	235	236	237	238
	242	241	243	240	242	245	249	249	249	249
173	248	246	247	245	243	244	246	245	245	245
	215	215	217	222	223	224	222	212	203	198
	109	109	104	104	103	101	101	110	116	115
176	180	175	173	169	172	176	180	181	184	191
	250	252	250	250	248	250	250	248	251	252
	242	242	239	238	239	243	243	247	248	248
179	250	252	252	252	253	253	252	254	253	253

Figure S12. Experimental input and output comparison in edge detection of frame 99 shown in Fig. 4c. (a-b) The input frame. (c-d) The output frame after the edge detection operation using our analog tensor core. An area of 11 × 11 pixels of the input frame is used for the comparison, and the corresponding output frame is presented in 10 × 10 pixels. The values presented are the grayscale values of each pixel, i.e. pixel intensity, I_n. The red lines in (c-d) enclose the vertical and horizontal edges detected in the output frame, respectively, and those in (a-b) mark the corresponding edges in the vertical and horizontal direction the input frame, respectively. Mean contrast in pixel intensity is introduced to evaluate the edge detection,

$$Contrast = \frac{1}{N}\sum_{n=1}^{N}(I_\mathrm{n,left} - I_\mathrm{n,right}) \text{ for vertical edge detection,}$$

$$\text{or alternatively, } Contrast = \frac{1}{N}\sum_{n=1}^{N}(I_\mathrm{n,up} - I_\mathrm{n,bottom}) \text{ for horizontal edge detection,}$$

where the differences in I_n of each pixel on both sides of the edges are averaged and evaluated. A larger mean contrast means that the edges are more determined from the neighboring pixels. As estimated, the mean contrast of the input frame is 81.27 for (a) and 55.18 for (b), while the mean contrast of the corresponding output frame is 116.17 for (c) and 80.95 for (d). The substantial increments in the mean contrast of (c-d) indicate that the edges are well detected by our analog tensor core.

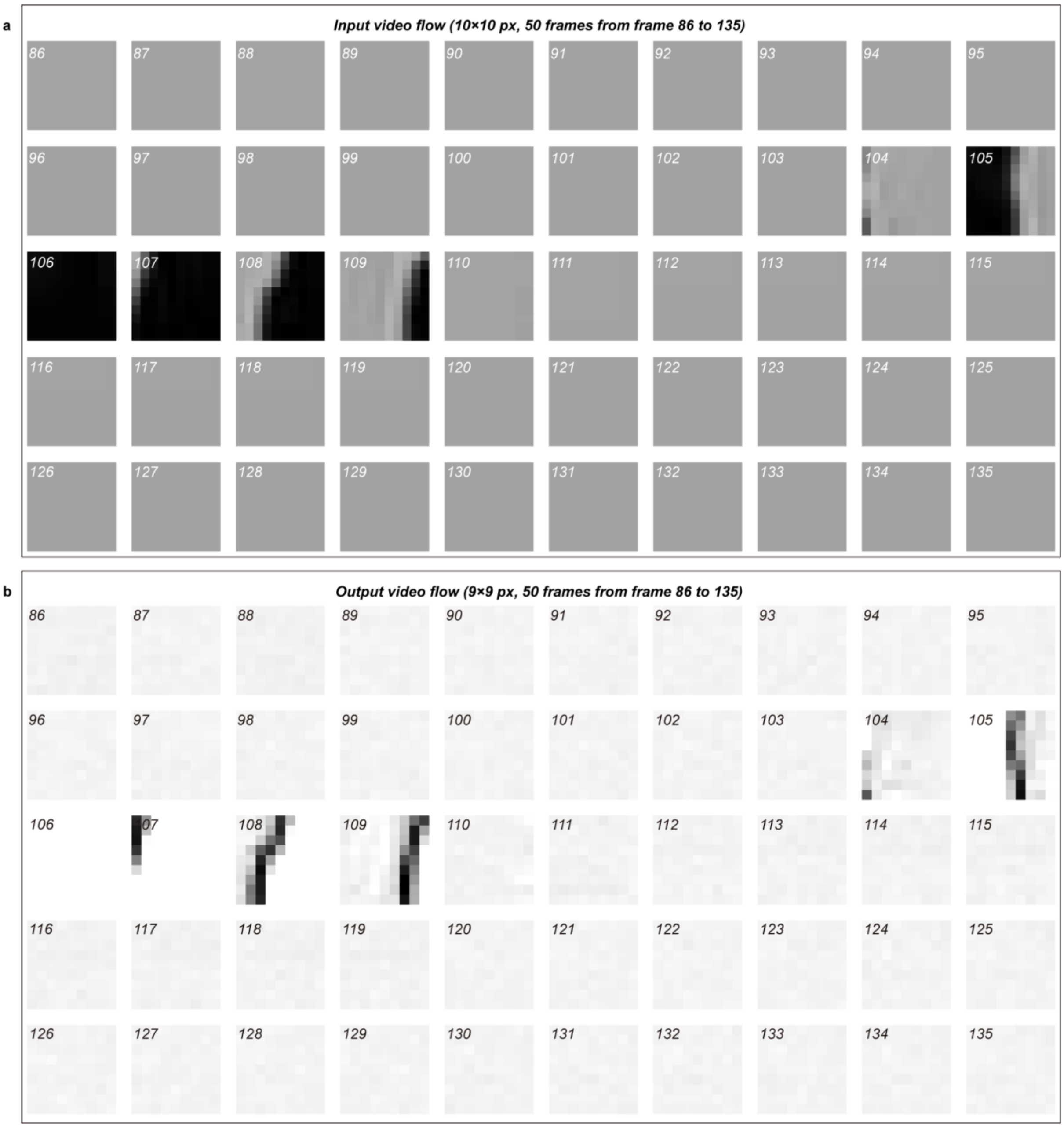


Figure S13. Visualized experimental edge detection of the video flow shown in Fig. 4e. (a) Input and (b) the reconstructed edge detection output of the video flow frame 86 to 135. The edges are clearly detected when the object passes during frame 105 to 109. See Movie S2 for the edge detection.

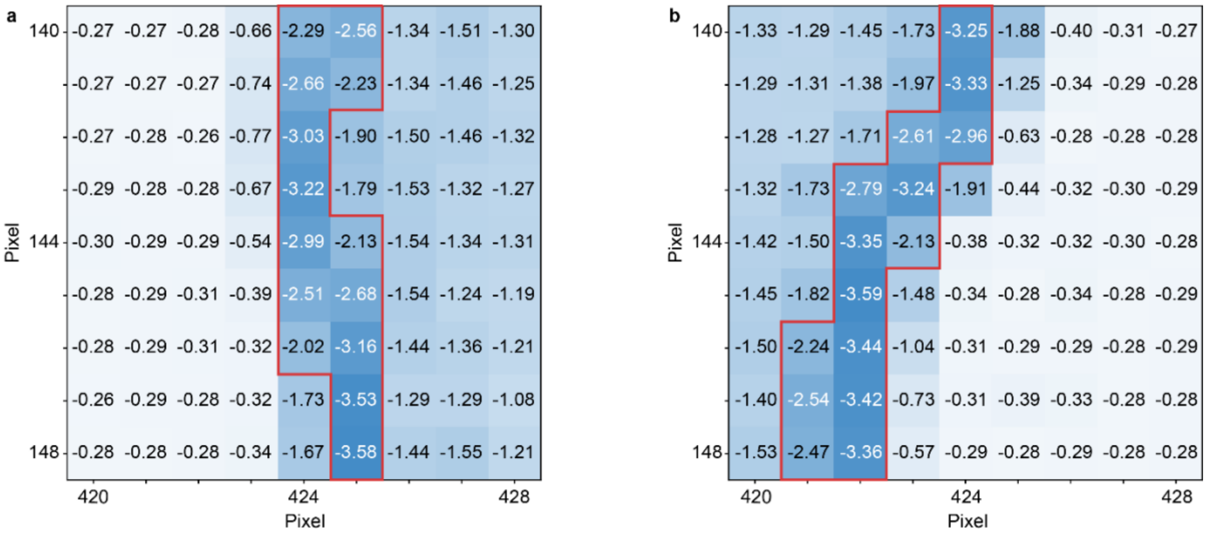
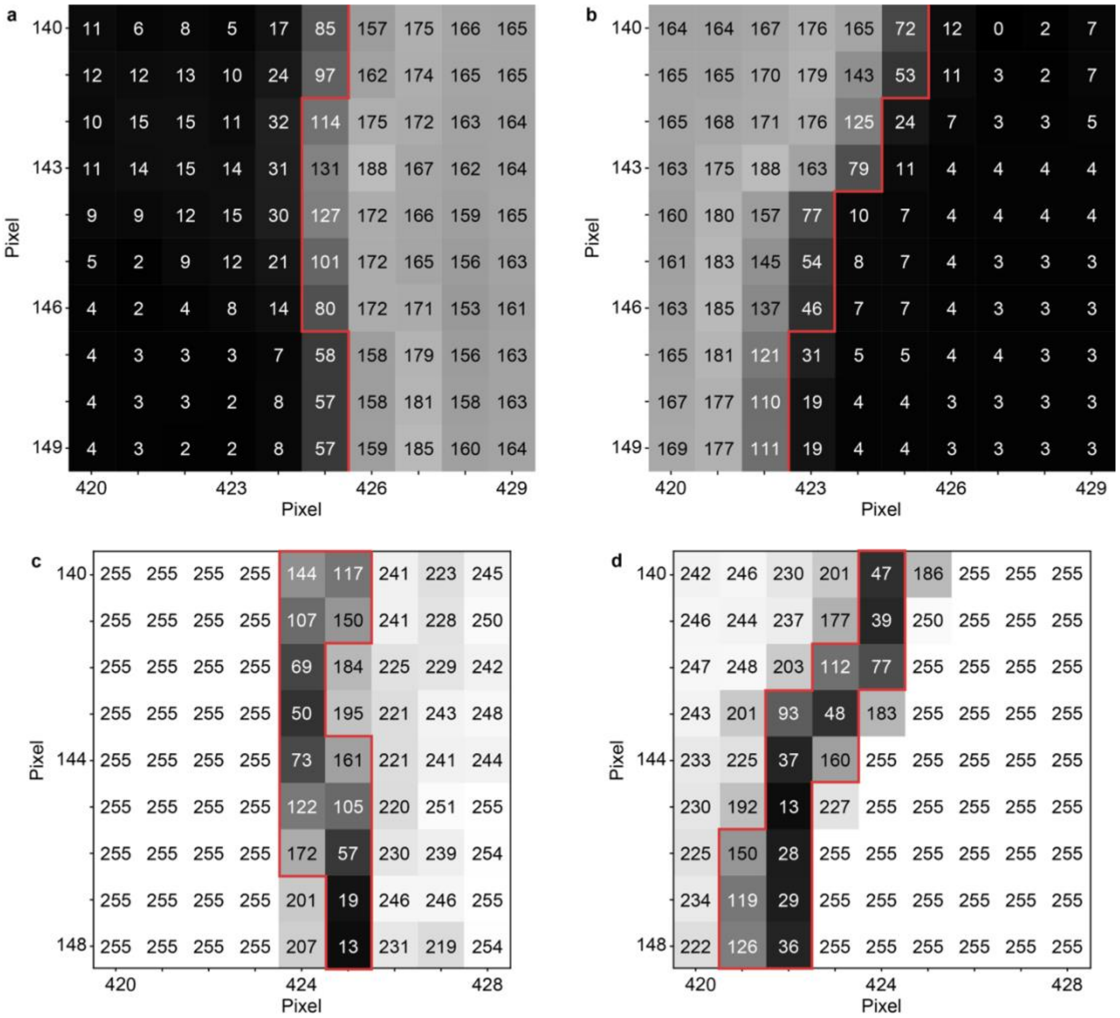


Figure S14. Experimental real-time edge detection output heatmaps of frame 105 (a) and frame 108 (b) shown in Fig. 4e. An area of 9 × 9 pixels is used to evaluate the real-time edge detection from 10 × 10 pixels input using our analog tensor core. The values presented in the heatmaps are the output voltages (V_{out}) from the tensor core, and the edges detected are enclosed by the red outlines. The SNR is estimated as 10.76 dB for frame 105 (a) and 13.55 dB for frame 108 (b), respectively. The dynamic range is estimated as 22.78 dB for frame 105 (a) and 22.47 dB for frame 108 (b), respectively. This proves the ability of our analog tensor core in performing efficient, low-pression real-time edge detection scenarios at edge.



a

	420			423			426			429
140	11	6	8	5	17	85	157	175	166	165
	12	12	13	10	24	97	162	174	165	165
	10	15	15	11	32	114	175	172	163	164
143	11	14	15	14	31	131	188	167	162	164
	9	9	12	15	30	127	172	166	159	165
	5	2	9	12	21	101	172	165	156	163
146	4	2	4	8	14	80	172	171	153	161
	4	3	3	3	7	58	158	179	156	163
	4	3	3	2	8	57	158	181	158	163
149	4	3	2	2	8	57	159	185	160	164

b

	420			423			426			429
140	164	164	167	176	165	72	12	0	2	7
	165	165	170	179	143	53	11	3	2	7
	165	168	171	176	125	24	7	3	3	5
143	163	175	188	163	79	11	4	4	4	4
	160	180	157	77	10	7	4	4	4	4
	161	183	145	54	8	7	4	3	3	3
146	163	185	137	46	7	7	4	3	3	3
	165	181	121	31	5	5	4	4	3	3
	167	177	110	19	4	4	3	3	3	3
149	169	177	111	19	4	4	3	3	3	3

c

	420				424				428
140	255	255	255	255	144	117	241	223	245
	255	255	255	255	107	150	241	228	250
	255	255	255	255	69	184	225	229	242
	255	255	255	255	50	195	221	243	248
144	255	255	255	255	73	161	221	241	244
	255	255	255	255	122	105	220	251	255
	255	255	255	255	172	57	230	239	254
	255	255	255	255	201	19	246	246	255
148	255	255	255	255	207	13	231	219	254

d

	420				424				428
140	242	246	230	201	47	186	255	255	255
	246	244	237	177	39	250	255	255	255
	247	248	203	112	77	255	255	255	255
	243	201	93	48	183	255	255	255	255
144	233	225	37	160	255	255	255	255	255
	230	192	13	227	255	255	255	255	255
	225	150	28	255	255	255	255	255	255
	234	119	29	255	255	255	255	255	255
148	222	126	36	255	255	255	255	255	255

Figure S15. Experimental input and output comparison in real-time edge detection of frame 105 and frame 108 shown in Fig. 4e. (a-b) The input frames. (c-d) The output frames after real-time edge detection operation using our analog tensor core. An area of 10 × 10 pixels of the input frames is used for the comparison, and the corresponding output frames are presented in 9 × 9 pixels. The values presented are the grayscale values of each pixel, i.e. pixel intensity, I_{n}. The red lines in (c-d) enclose the edges detected in the output frames, respectively, and those in (a-b) mark the corresponding edges in the input frames, respectively. As estimated, the mean contrast of the input frames is 86.5 for frame 105 (a) and 69.6 for frame 108 (b), while the mean contrast of the corresponding output frame is 149.56 for frame 105 (c) and 160.44 for frame 108 (d). The substantial increments in the mean contrast of (c-d) indicate that the edges are well detected by our analog tensor core in real-time.

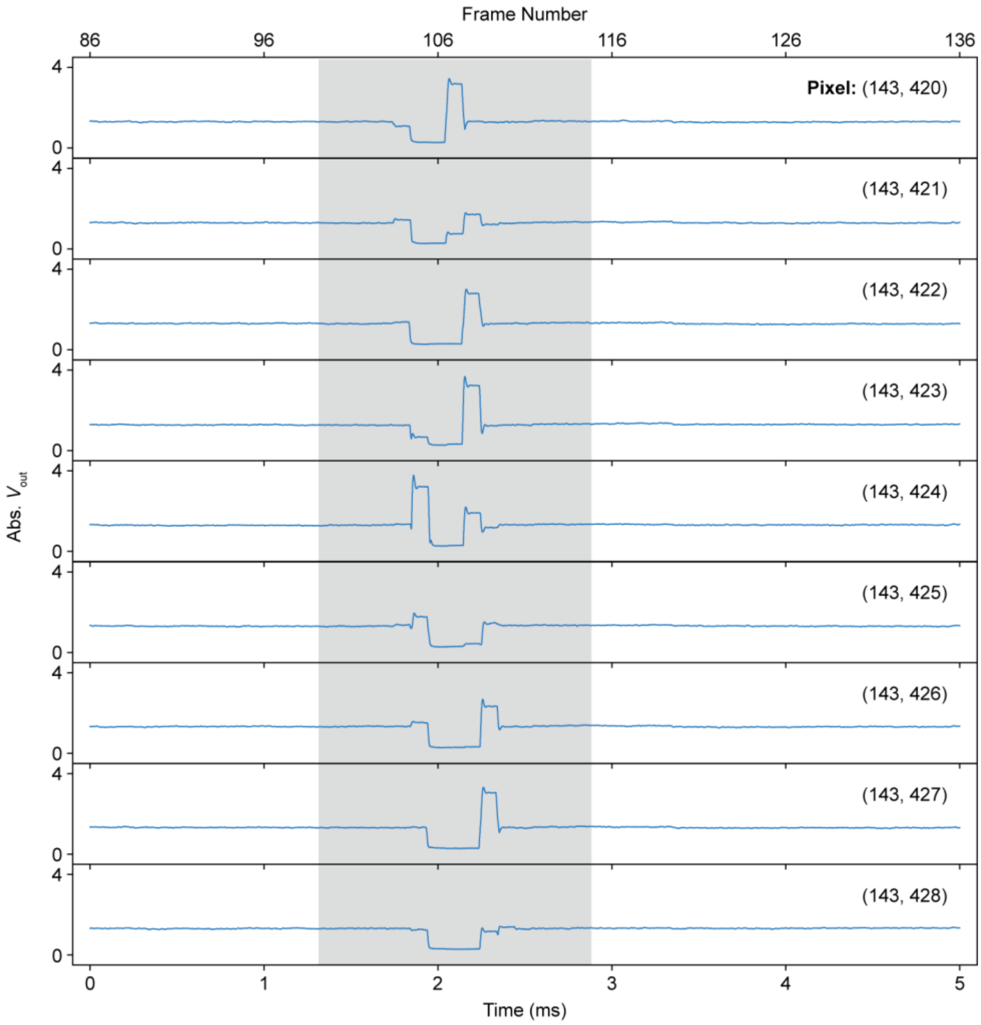


Figure S16. Experimental edge detection outputs of 9 selected pixels in the video flow shown in Fig. 4f. The peaks represent the detected edges of the corresponding pixels in the video flow. The coordinates represent the selected pixels. The results here show the first 50 frames of the video flow. Each frame is processed in a time scale of 0.1 ms. The output voltages are sampled according to the input frequency intervals, and are then normalized and reconstructed to output the pixel intensity values. The gray region corresponds to the regions presented in Fig. 4f.

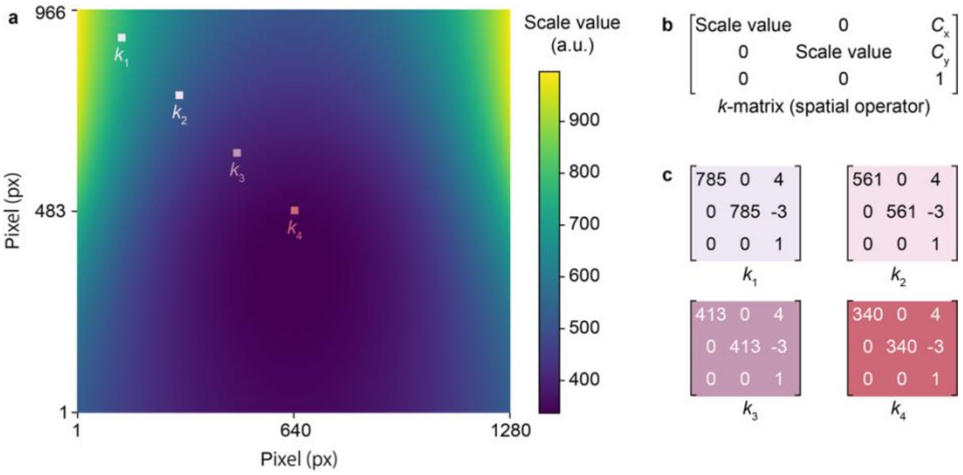


Figure S17. Visualized k-matrix (i.e. matrix of the spatial operators) of the viewfield distortion correction task shown in Fig. 5a. (a) Mapping of the scale values of the k-matrix across the viewfield. (b) The general formular of the k-matrix, where the scale value variates at the specific pixel across the viewfield, and the C_x and C_y are the constants corresponding to the fisheye lens configuration. Across the k-matrix, the scale value is solved by the distortion correction arithmetic as reported in ref. 3. C_x and C_y are 4 and -3, respectively, according to the configuration of the fisheye lens. (c) Four specific k-matrix units (i.e. spatial operators) at four specific locations across the viewfield. The four locations are shown in (a).

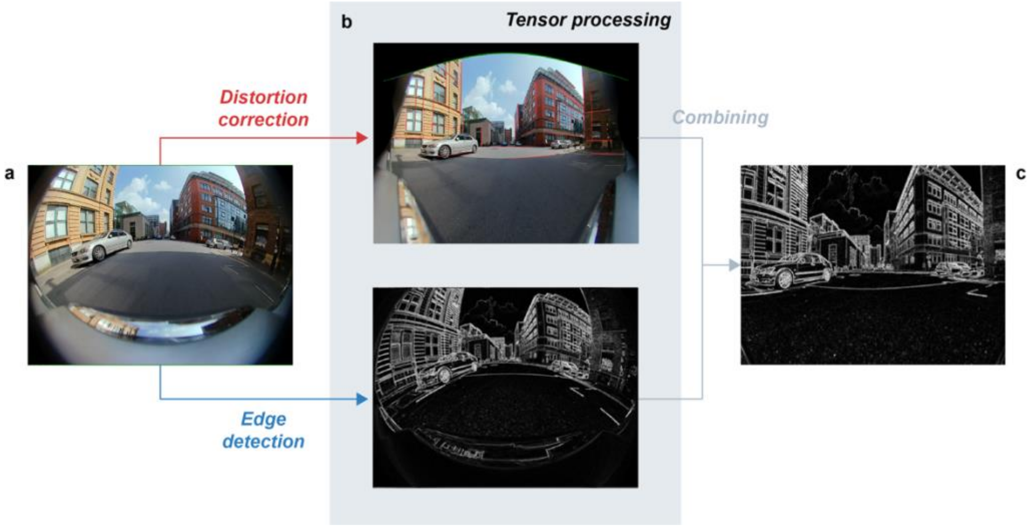


Figure S18. Tensor processing workflow using our CNT tensor core. The input image as shown in (a) is processed in (b) parallel tensor processing to perform the distortion and edge detection tasks. The images after the parallel distortion and edge detection operations are then combined to render the viewfield-corrected edge detection image shown in (c). The combined result is cropped at the edges for the convenience of viewfield-correction representation.

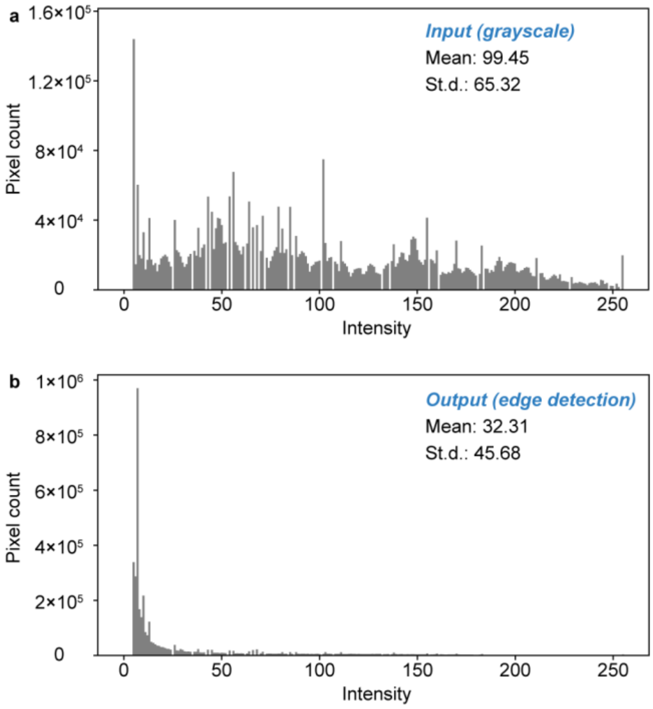


Figure S19. Distribution of pixel intensity in Fig. 5b. The distribution of pixel intensity in (a) the fisheye capture input and (b) the edge detected output. The standard deviation of the pixel intensity decreases significantly from 65.32 to 45.68, proving the effectiveness of the edge detection using our tensor core.

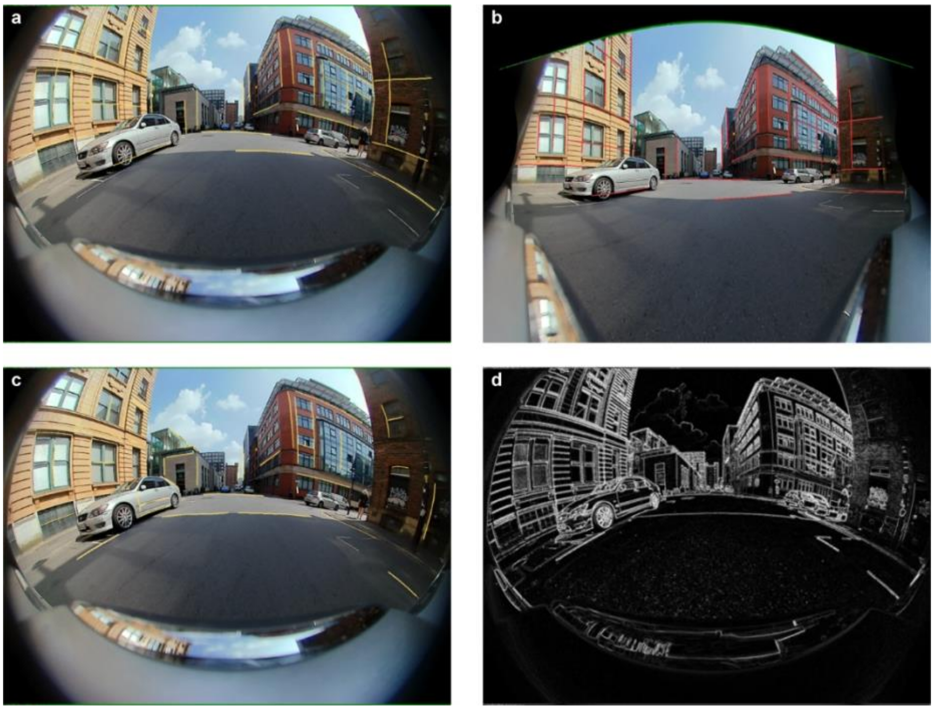


Figure S20. Selected areas for quantifying the parallel tensor processing shown in Fig. 5e and f. The 36 areas of the (a) fisheye capture are marked with yellow lines for quantifying the (b) distortion correction and (c) edge detection, respectively. The corresponding results are marked with red lines for distortion correction in (b), and grayscale lines for edge detection in (d). To quantify the distortion correction, we select the regions that are expected to be straight lines in the real world. For each region, we apply the least squares method to fit the set of points before and after correction, and then measure its deviation from the ideal straight line. The deviation is quantified as the mean absolute error (MAE), defined as the mean of the perpendicular distances from each point to the fitted line:

$$d_i = \frac{|kx_i - y_i + b|}{\sqrt{k^2+1}},$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N} d_i,$$

where (x_i, y_i) represents the coordinates of each point, and k and b are the slope and intercept of the fitted line, respectively. By comparing the MAE values before and after correction, we can quantitatively assess the improvement in the straightness and thus, the effectiveness of the distortion correction. The mean MAE of the 36 areas decreases from 2.84 to 0.45 (Fig. 5g), proving an efficient distortion correction. To quantify the edge detection, we select the regions with edges, and calculate the mean contrast via the method described in Fig. S12. Note that, to minimize the impact of the inaccurate manual line placement and complex surrounding edges on the accuracy of the quantitative results, we calculate the edge strength as the mean intensity across a three-pixel-wide line centered on the selected edge segment (i.e., averaging one pixel on each side of the edge). For the non-edge regions, we average the intensities of five pixels on each side of the edge segment. This approach enables a more accurate measurement of mean contrast across the edge. The effectiveness of edge detection can then be quantitatively assessed by the observed increase in contrast (from 32.67 to 66.22) after edge detection.

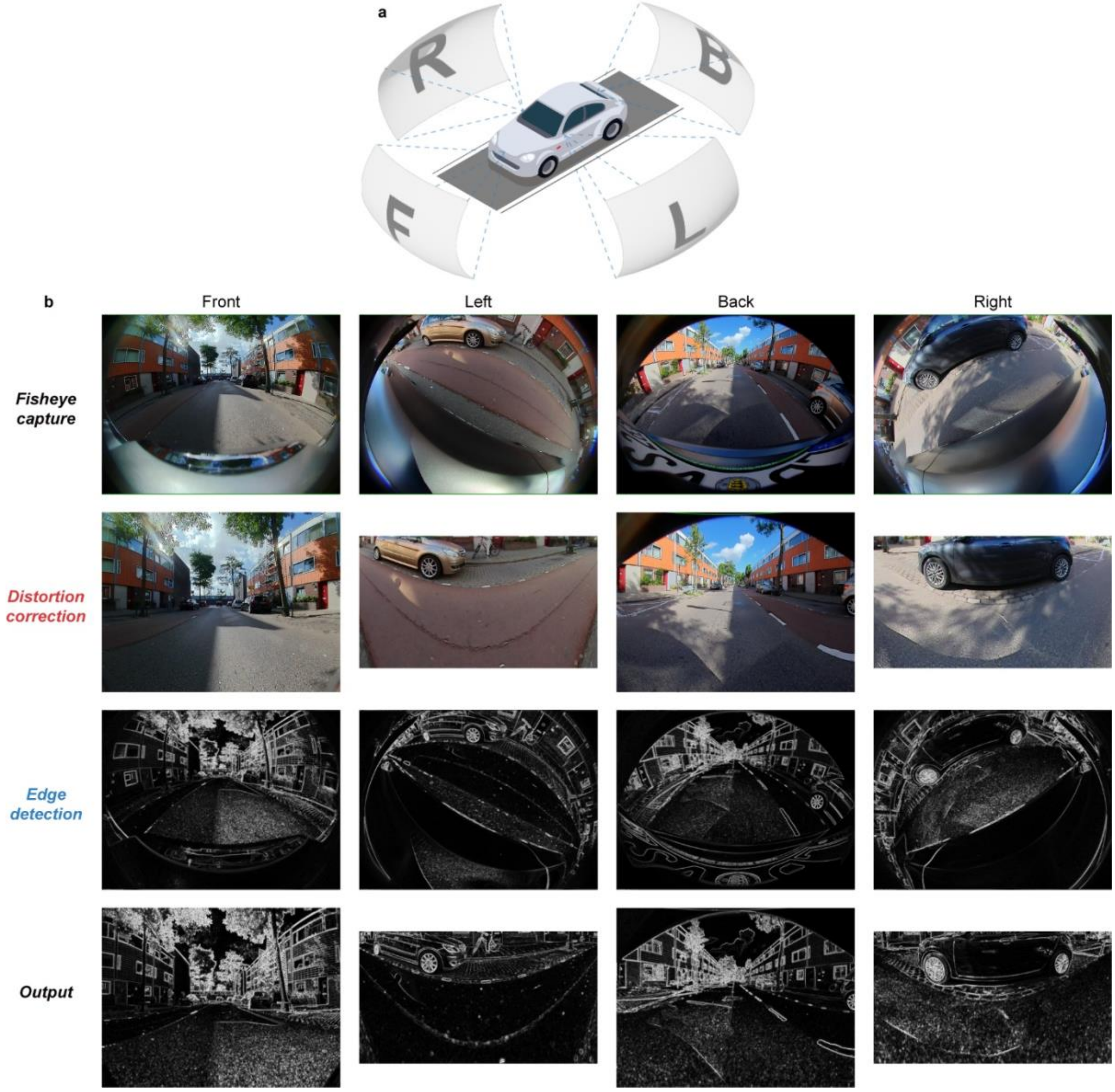


Figure S21. Large-scale image processing in autonomous driving using our CNT analog tensor core via simulation. (a) Schematic omnidirectional perception in autonomous driving. The front, left, back, and right sides of visual information are captured from the corresponding fisheye cameras. (b) Omnidirectional visual information perception, accomplishing parallel distortion correction and edge detection tasks using our analog tensor core. The outputs from distortion correction and edge detection are subsequently combined to output the viewfield-corrected edge detection images. Note that, as the parameters of the fisheye cameras are different in the front, left, back, and right four sides, the *k*-matrixes in each sides are adjusted accordingly in the distortion correction task.[4]

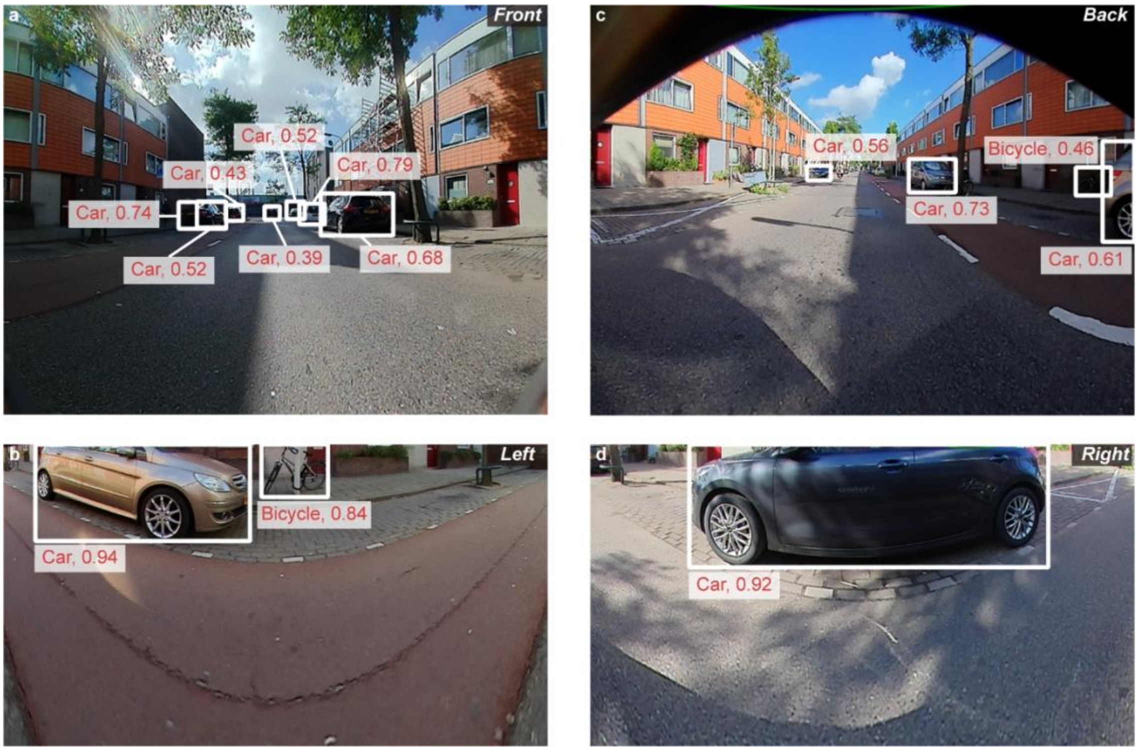


Figure S22. Object recognition after the large-scale tensor processing using our CNT analog tensor core. The object recognition results in the (a) front, (b) left, (c) back, and (d) right view of the street after the distortion correction and edge detection tensor processing. The cars and bicycles are accurately recognized with high confidence, proving the effectiveness of performing tensor processing using our analog tensor core.

Table S1. Error evaluation of spatial transformation in Fig. 3b-c, (2). To quantitively evaluate the identical spatial transformation (i.e. 0° rotation around the x- y- and z-axes) of the 5 selected points in Fig. 3b-c, (2), we present in the table the coordinates of the inputs and outputs, respectively, and the estimated percentage error (PE) and relative *Euclidean* error (REE). PE is estimated by $PE = |c_{\text{output}} - c_{\text{input}}|/|c_{\text{input}}| \times 100\%$, where c represents the coordinates of the inputs and outputs along the x- y- and z-axes. RPE is estimated by $REE = D/A_{\text{input}} \times 100\%$, where $A_{\,input}$ is the magnitude of the coordinate vector, and D is the *Euclidean* distance, $D = \sqrt{(x_{output} - x_{input})^2 + (y_{output} - y_{input})^2 + (z_{output} - z_{input})^2}$. PE evaluates the spatial transformation error at an individual coordinate along the x- y- and z-axes, and REE evaluates the overall error of the spatial transformation. As estimated, PE is <0.73% and REE is <1.12% for the 5 selected points. PE and REE for the 867 coordinates of the 3D model are <1.47% and <2.81%, respectively, proving the high accuracy spatial transformation performed using our CNT tensor core.

Points	*Input*			*Output*			*PE*			*REE*
	x	y	z	x’	y’	z’	x-axis	y-axis	z-axis	–
1	0.160	0.700	0.715	0.163	0.696	0.713	0.42%	0.42%	0.15%	0.61%
2	0.097	0.786	0.181	0.096	0.781	0.188	0.04%	0.55%	0.73%	1.12%
3	0.198	0.948	0.442	0.200	0.944	0.442	0.14%	0.40%	0.00%	0.40%
4	0.440	0.284	0.627	0.437	0.278	0.623	0.23%	0.52%	0.41%	0.87%
5	0.182	0.195	0.324	0.183	0.197	0.322	0.13%	0.14%	0.21%	0.67%

Table S2. Error evaluation of spatial transformation in Fig. 3b-c, (3). To quantitively evaluate the 90° rotation around the x-axis of the 5 selected points in Fig. 3b-c, (3), we present i) the calculated (cal.) values for the expected coordinates of x-axis 90° rotation, ii) the output values for the remapped coordinates from the experimental results, and iii) the estimated PE and REE in the table. As estimated, PE is <0.68% and REE is <1.06% for the 5 selected points. PE and REE for the 867 coordinates of the 3D model are <1.48% and <2.39%, respectively, proving the high accuracy spatial transformation performed using our CNT tensor core.

Points	*Cal. x-axis 90^o rotation*			*Output x-axis 90^o rotation*			*PE*			*REE*
	x	y	z	x’	y’	z’	x-axis	y-axis	z-axis	–
1	0.160	0.285	0.700	0.161	0.287	0.693	0.07%	0.20%	0.68%	0.92%
2	0.097	0.819	0.786	0.094	0.814	0.784	0.29%	0.57%	0.28%	0.61%
3	0.198	0.558	0.948	0.197	0.557	0.946	0.17%	0.16%	0.24%	0.30%
4	0.440	0.373	0.284	0.435	0.376	0.279	0.44%	0.26%	0.45%	1.06%
5	0.182	0.676	0.195	0.180	0.678	0.194	0.19%	0.15%	0.09%	0.36%

Supplementary References

1. Pei, J. *et al.* Scalable Synaptic Transistor Memory from Solution-Processed Carbon Nanotubes for High-Speed Neuromorphic Data Processing. *Adv Mater* **37**, e2312783 (2025).
2. Gu, J. *et al.* Solution-Processable High-Purity Semiconducting SWCNTs for Large-Area Fabrication of High-Performance Thin-Film Transistors. *Small* **12**, 4993–4999 (2016).
3. Weng, J., Cohen, P. & Herniou, M. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 965–980 (1992).
4. Yogamani, S. *et al.* WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving. in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 9307–9317 (2019).