between an event stack (Eq. (4)) and the two consecutive coded-aperture images (Eq. (1)) as

$$E^{(n,n+1)}_{x,y} \approx \frac{\log\left(I^{(n+1)}_{x,y}\right) - \log\left(I^{(n)}_{x,y}\right)}{\tau}. \tag{6}$$

Assuming that equality holds for Eq. (6) and combining it with Eq. (3), we can obtain $I^{(1)}, \dots, I^{(N)}$ as

$$I^{(n)}_{x,y} = I^{(1)}_{x,y} \exp\left(\tau \textstyle\sum_{2\le k\le n} E^{(k-1,k)}_{x,y}\right) \; (n \ge 2) \tag{7}$$

$$I^{(1)}_{x,y} = \frac{\bar{I}_{x,y}}{1+\sum_{2\le n\le N}\exp\left(\tau\sum_{2\le k\le n}E^{(k-1,k)}_{x,y}\right)}. \tag{8}$$

This means that under the equality assumption for Eq. (6), N coded-aperture images ($I^{(1)}, \dots, I^{(N)}$) can be derived analytically from the data observed with our imaging method ($\bar{I}$ and $E^{(1,2)}, \dots, E^{(N-1,N)}$). Therefore, we can state that our imaging method is *quasi-equivalent* to the baseline coded-aperture imaging method.

Interestingly, this type of quasi-equivalence does not hold for joint aperture-exposure coding. By using Eq. (1), the imaging process of Eq. (2) is rewritten as

$$I_{x,y} = \sum_n p^{(n)}_{x,y} \, I^{(n)}_{x,y}. \tag{9}$$

Obviously, it is impossible to analytically obtain N coded-aperture images ($I^{(1)}, \dots, I^{(N)}$) from a single observed image I alone. Therefore, our imaging method has a theoretical advantage over joint aperture-exposure coding. However, this theory alone is insufficient to ensure the practicality of our method. The actual event data are very noisy and harshly quantized (by the contrast threshold τ), which breaks the equality assumption for Eq. (6).

3.3. Algorithm

We developed an end-to-end trainable algorithm on the basis of deep-optics [13, 14, 21, 28, 33, 48, 52, 54], in which the camera-side optical-coding patterns and the light-field reconstruction algorithm were jointly optimized in a deep-learning-based framework. We carefully designed each part of our algorithm to ensure the compatibility with real camera hardware. Although we specifically mention the hardware setup that is available to us, the ideas behind our design would be useful for other possible hardware setups. We set $N = 4$ unless otherwise mentioned.

Our algorithm consists of two parts: AcqNet and RecNet. AcqNet describes the data-acquisition process using a coded aperture and an event camera as

$$\bar{I}, E^{(1,2)}, E^{(2,3)}, E^{(3,4)} = \text{AcqNet}(L). \tag{10}$$

Algorithm 1 Pseudo-code for AcqNet

```
1: trainable tensors: $\alpha, \beta \in \mathcal{R}^{8\times 8}$
2: forward($L$):
3:     set $s$
4:     $a^{(1)}, a^{(3)} = \text{sigmoid}(s\alpha), \text{sigmoid}(s\beta)$
5:     $a^{(2)}, a^{(4)} = 1 - a^{(1)}, 1 - a^{(3)}$
6:     for $n$ in [1, 2, 3, 4]:
7:         compute $I^{(n)}$ by Eq. (1)
8:         if $n > 1$:
9:             compute $E^{(n-1,n)}$ by Eq. (12)
10:        end
11:    end
12:    compute $\bar{I}$ by Eq. (3)
13:    return $\bar{I}, E^{(1,2)}, E^{(2,3)}, E^{(3,4)}$
```

The trainable parameters of AcqNet are related to the aperture's coding patterns. RecNet receives the observed data as the input and reconstructs the original light field as

$$\hat{L} = \text{RecNet}(\bar{I}, E^{(1,2)}, E^{(2,3)}, E^{(3,4)}). \tag{11}$$

AcqNet and RecNet are jointly trained to minimize the reconstruction (MSE) loss between L and $\hat{L}$. Once the training is finished, AcqNet is replaced with the physical imaging process of the camera hardware, in which the coding patterns are adjusted to the learned parameters of AcqNet. The data acquired from the camera are fed to RecNet to reconstruct the light field of a real 3-D scene.

Hardware-driven constraints for coded aperture. Similar to some previous studies [14, 26, 30], we used a liquid-crystal-on-silicon (LCoS) display (Forth Dimension Displays, SXGA-3DM) to implement a coded aperture. This display can output a sequence of semi-transparent coding patterns repeatedly. We need to consider the following two constraints. **Binary constraint**. Although our LCoS display can support both binary and grayscale patterns, a grayscale pattern is actually represented as a temporal series of multiple binary patterns; a grayscale transmittance is represented as the ratio of 0/1 periods. To avoid unintended 0/1 flips, we choose to use only binary patterns for aperture coding. **Complementary constraint**. Our LCoS display requires a "DC balance"; a certain pattern a and its complement $a^* = 1 - a$ should be included in the sequence. Since the events are recorded continuously over time, we use both a and a^* as the coding patterns.

AcqNet. A pseudo-code of AcqNet is presented in Algorithm 1. In lines 4 and 5, we make the coding patterns compatible with the binary and complementary constraints. More specifically, we prepare two sets of trainable tensors, each with 8 × 8 elements, denoted as α and β. They are multiplied by the scale parameter s then fed to the sigmoid function to produce the coding patterns $a^{(1)}$ and $a^{(3)}$. As the training proceeds, s gradually increases, which forces $a^{(1)}$ and $a^{(3)}$ to gradually converge to binary patterns. Moreover, $a^{(2)}$ and $a^{(4)}$ are made to be complementary to $a^{(1)}$