Supplementary information

Bioinspired planar intelligent nanophotonic sensor for wide-angle accurate motion perception and prediction

Zhongyi Yuan[1,3], Chengyao Hao[1,3], Yin Wang[2,3], Yue Wu[1], Muyang Li[1], Chengyi Feng[2], Zhenguo Hu[2], Zaichen Zhang[1,2*], Ji Chen[1,2*]

1. National Mobile Communications Research Laboratory, School of Information Science and Engineering, Frontiers Science Center for Mobile Information Communication and Security, Southeast University, Nanjing 210096, China.

2. Purple Mountain Laboratories, Nanjing 211111, China.

3. These authors contributed equally: Zhongyi Yuan, Chengyao Hao and Yin Wang.

*Corresponding author: Zaichen Zhang, Email: zczhang@seu.edu.cn;

Ji Chen, Email: jichen@seu.edu.cn.

This Supplementary information file includes:

Notes S1 to S17:

Table S1 to S5

Figures S1 to S14

Legends for Movies S1 to S2

Other Supplementary Materials for this manuscript include the following:

Movies S1 to S2

Note S1. Integration of PINS into a MAV platform.

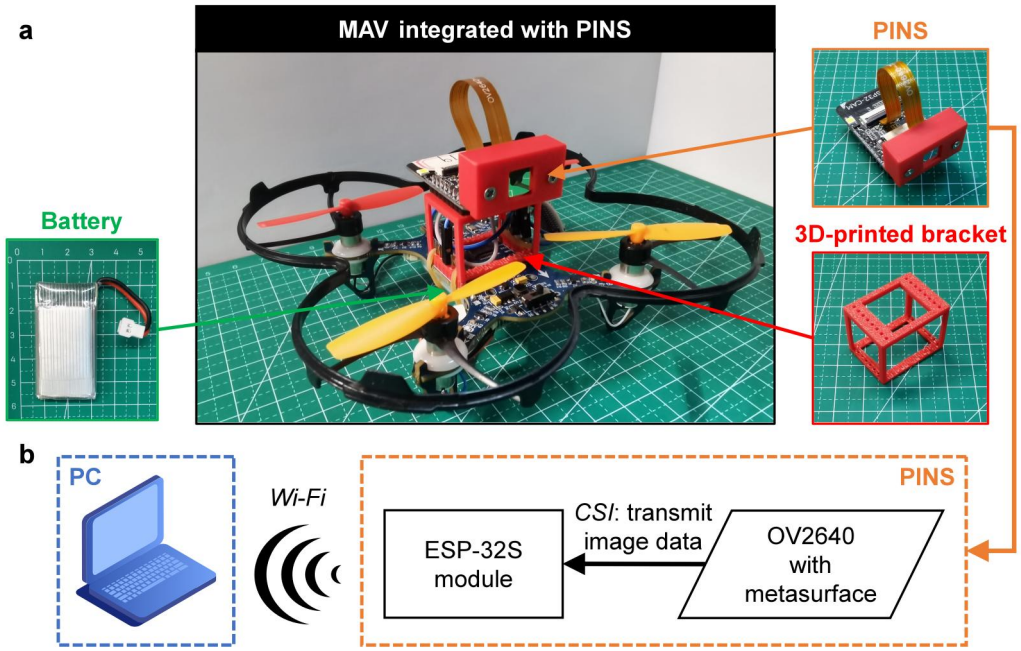

Figure S1 | Structural components of the PINS-integrated MAV system and its working principle for image data transmission. a, Structural components of the PINS-integrated MAV system. The system consists of three key components: the PINS module, a 5V lithium battery, and a 3D-printed mounting bracket. **b,** Schematic diagram illustrating the workflow of image data transmission.

Figure S1 illustrates the structural composition of the Planar Intelligent Nanophotonic Sensor (PINS) integrated miniature unmanned aerial vehicle (MAV) system, along with its working principle for image data transmission. As shown in **Fig. S1a**, the integrated system consists of three primary components: the PINS module, a 5V lithium battery, and a custom-designed 3D-printed mounting bracket. Stable and compact integration is achieved through precise mechanical interlocking between the bracket, the PINS module, and the MAV. In addition, elastic bands provide supplementary fixation, further enhancing the structural stability of the system.

As illustrated in **Fig. S1b**, the PINS module mounted on the MAV transmits real-time image data to a ground-based PC through a Wi-Fi link. Specifically, the CMOS sensor (OV2640)

captures the image formed by the metalens and transfers the image data to the microcontroller unit (MCU, ESP32-S module) through a flexible printed circuit (FPC) using the CSI image transmission protocol. After initialization, the ESP32-S establishes a Wi-Fi connection and launches an HTTP server, enabling the ground terminal to remotely access the real-time data stream through the assigned IP address. Benefiting from its integrated 2.4 GHz Wi-Fi capability and support for high-speed wireless communication (150 Mbps under the 802.11n protocol), the ESP32-S provides an efficient platform for real-time wireless image backhaul in the present system. Under typical operating conditions, the imaging module supports real-time transmission of UXGA-resolution images (1600 × 1200) at a frame rate of 15 frames per second.

Note S2. Ray-tracing verification of the focusing performance of metalenses with different FoV.

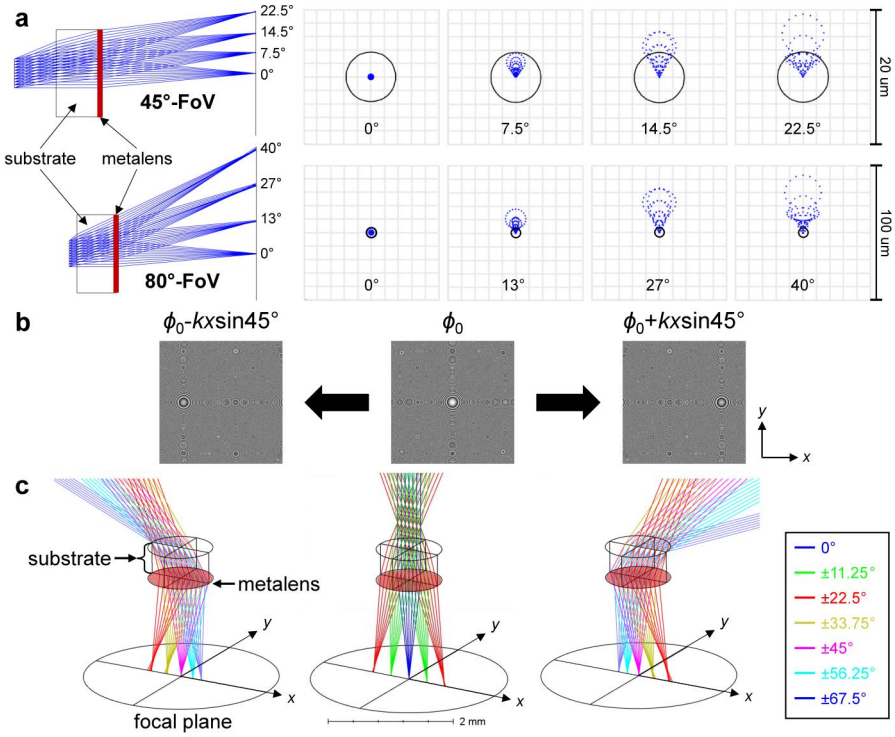

Figure S2 | Ray-tracing verification of metalenses with different FoV. a, Comparison of the ray-tracing results and focal spots for metalenses designed to cover different FoV. The black circles in the focal spot patterns are the Airy disks of the metalens. **b,** Phase profiles of the side-FoV metalenses are derived from the center-FoV metalens phase by introducing linear tilt terms of ±*kx*sin45°, respectively. **c,** Zemax ray-tracing simulations showing the focusing performance of the metalenses under different incident angles.

To determine a suitable field of view (FoV) for each individual metalens, we first carried out ray-tracing simulations in Zemax OpticStudio to evaluate and compare the focusing performance of metalenses designed for different angular ranges, as shown in **Fig. S2a**. In this comparison, we examined the focal spot distributions at several representative incident angles across the designed FoV and assessed their quality by comparing them with the corresponding Airy-disk sizes. The simulation results show that, when the FoV of a single metalens is designed to be 45°,

the focusing performance remains well preserved over the entire angular range considered. In particular, the focal spots at representative incident angles stay close to the Airy-disk size, indicating that the metalens can still provide high-quality focusing within this FoV. By contrast, when the target FoV of a single metalens is further enlarged to 80°, the focusing quality deteriorates noticeably, and the degradation becomes especially evident near the edge of the designed angular range. The focal spots under these larger incident angles become significantly broader and deviate more substantially from the Airy-disk limit. Based on this direct comparison, we selected 45° as the FoV of the central metalens.

We next verified that the acquisition FoV of an individual metalens can be shifted from the central viewing direction to a side angular sector through the introduction of an additional tilted phase term. Starting from the phase profile of the center-FoV metalens, the phase profiles of the two side-FoV metalenses were generated by superimposing linear tilt terms, following the design principle described in the main text, as illustrated in **Fig. S2b**. From a physical point of view, this additional phase term shifts the effective viewing direction of the metalens, so that incident rays from an off-axis angular range are redirected toward the near-axis focusing regime of the original metalens design and are then focused in a similar manner. In this way, the linear tilted phase does not simply alter the outgoing wavefront direction, but effectively transfers the usable focusing range of the metalens from the central FoV to a designated side FoV. The corresponding ray-tracing results for the side-FoV metalenses are presented in **Fig. S2c**. These simulations show that, after the tilted phase is introduced, the metalenses no longer exhibit their optimal focusing performance around the central viewing direction, but instead within the shifted off-axis angular range. The focal behavior therefore follows the designed change in viewing direction, confirming that the FoV can indeed be reassigned through the tilted-phase modulation.

At the same time, the focusing performance within the shifted angular range remains acceptable and comparable to that of the central-FoV metalens over its own designated range. These results verify the validity of the FoV-shifting mechanism used in this work. Therefore, by combining three metalenses, each individually designed to cover a 45° FoV but assigned different viewing directions, the metalens array can jointly provide a total wide-angle imaging FoV of 135°.

Note S3. Wide-angle image reconstruction pipeline.

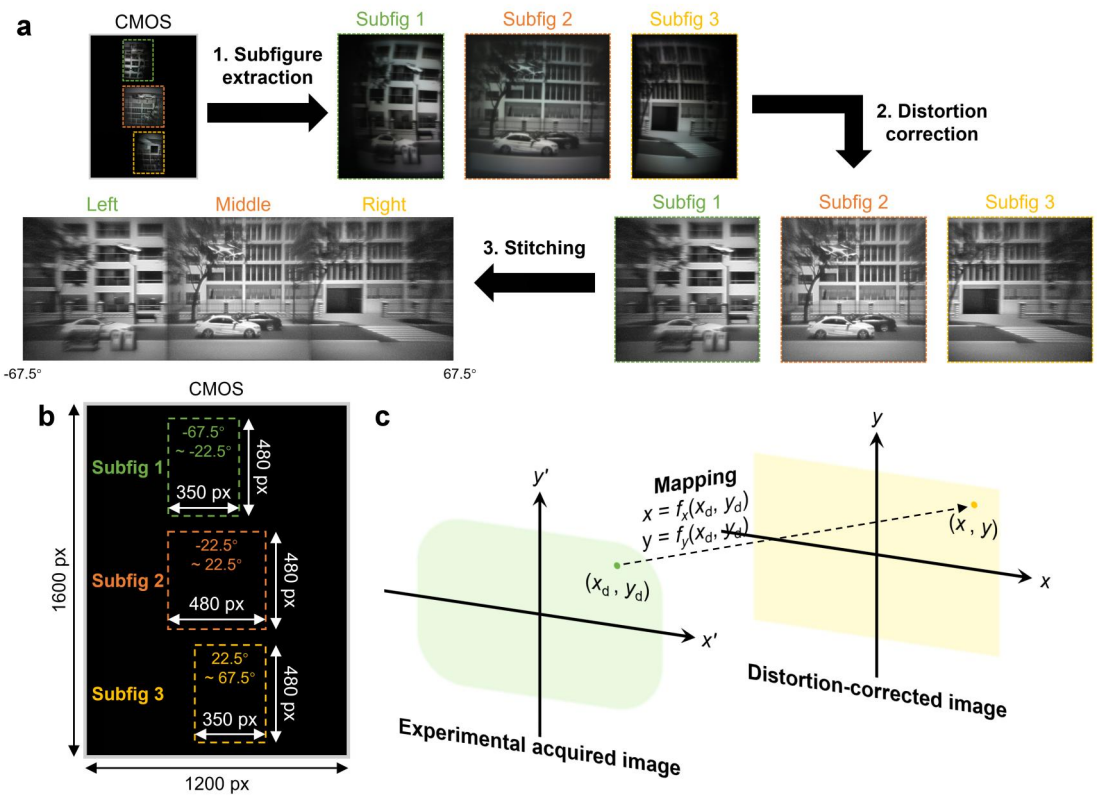

Figure S3 | Wide-angle image reconstruction pipeline for the metalens array. a, Reconstruction process of the wide-angle image, including sub-image extraction from the raw CMOS image, distortion correction of each sub-image, and final image stitching. **b,** Layout of the three sub-images on the CMOS sensor. The dashed boxes indicate the spatial regions occupied by the sub-images formed by the left, middle, and right metalenses, respectively. **c,** Schematic illustration of the distortion correction process, where the distorted pixel coordinates in each sub-image are mapped to corrected coordinates in the rectified image.

To reconstruct a wide-angle image from the raw output of the metalens array, we developed a pipeline consisting of three sequential steps: sub-image extraction, distortion correction, and image stitching, as illustrated in **Fig. S3a**.

The spatial layout of the sub-images on the CMOS sensor is determined by both the sensor geometry and the requirement to avoid crosstalk between adjacent imaging channels. The CMOS

sensor (OV2640, Omnivision) has a resolution of 1600 × 1200 pixels. To maximize utilization of the available sensor area while preventing overlap between the images formed by different metalenses, the three sub-images are arranged in a staggered configuration on the CMOS plane, as shown in **Fig. S3b**. In this configuration, each metalens projects its image onto a distinct and non-overlapping region, which suppresses inter-channel crosstalk and facilitates subsequent image processing. Based on this layout, the region of interest corresponding to each metalens is first extracted from the raw CMOS image. Since the three metalenses are designed for different viewing directions, the extracted sub-images exhibit different imaging distortions and cannot be stitched directly in their raw form. Therefore, each cropped sub-image is individually rectified before stitching.

The distortion correction is accomplished by establishing a mapping between distorted image coordinates and corrected image coordinates, as illustrated in **Fig. S3c**. To account for both the asymmetric distortion for the side-FoV metalenses and symmetric distortion for the central-FoV metalens, we employ a polynomial fitting approach that establishes an accurate mapping between object points (x, y) and distorted image points (x_d, y_d). For each metalens, we capture images of a uniform grid of points, fit the polynomial coefficients f_x and f_y to establish this mapping, and then apply it to correct every pixel in the captured image. After distortion correction, the three processed sub-images represent contiguous angular segments of the full scene. These rectified sub-images are then concatenated horizontally to generate the final wide-angle reconstructed image (480 × 1440 pixels), as shown in **Fig. S3a**.

Note S4. Model of sensor noise during PSF-based metalens imaging simulation.

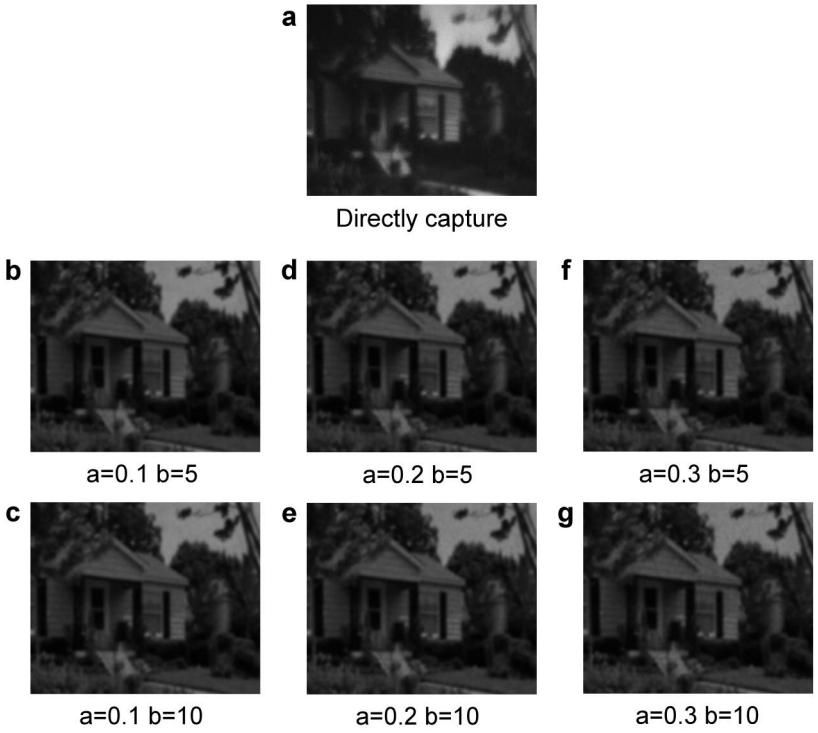

Figure S4 | Comparison of sensor noise levels during PSF-based metalens imaging simulation. a, The image captured directly by the CMOS sensor. **b-c,** Simulated imaging results when parameter a = 0.1 and b is set to 5 and 10, respectively. **d-e,** Simulated imaging results when parameter a = 0.2 and b is set to 5 and 10, respectively. **f-g,** Simulated imaging results when parameter a = 0.3 and b is set to 5 and 10, respectively.

In the PSF-based metalens imaging simulation process, sensor noise arises during CMOS imaging and primarily originates from two sources: photon shot noise and thermal read noise. Photon shot noise is caused by fluctuations in photon arrival due to uneven illumination and inherent temporal randomness. It follows a Poisson distribution, where the number of photons received at each pixel is modelled as a random variable:

$$P\left(k_{\mathrm{ph}}, \lambda_{\mathrm{br}}\right) = \frac{\lambda_{\mathrm{br}}^{k_{\mathrm{ph}}} \times e^{-\lambda_{\mathrm{br}}}}{k_{\mathrm{ph}}!} \tag{S1}$$

Here, λ_{br} is the expected number of photon arrivals, which increases with the image brightness, and k_{ph} is the actual number of photons received. Photon shot noise becomes more significant in low-light images.

Thermal noise is the noise generated by the sensor during signal readout process. It is associated to the sensor's temperature and the operating state of the electronic components. Typically, thermal noise follows a zero-mean Gaussian distribution.

To facilitate the noise modeling, we assume that the noisy image **Y** can be decomposed into the following formula:

$$\mathbf{Y}(\mathbf{r}) = y(\mathbf{r}) + \eta_{\text{photon}}\left[y(\mathbf{r})\right] + \eta_{\text{thermal}}(\mathbf{r}) \tag{S2}$$

where $\eta_{\text{photon}}\left[y(\mathbf{r})\right]$ represents photon shot noise, dependent on local pixel value $y(\mathbf{r})$, and $\eta_{\text{thermal}}(\mathbf{r})$ represents brightness-independent thermal noise. By combining both components, the total noise model can be formulated as:

$$z(\mathbf{r}) \quad N\left(\mu = y(\mathbf{r}), \sigma^2 = a y(\mathbf{r}) + b\right) \tag{S3}$$

where, μ represents the original pixel value of the image, σ^2 is the variance of the noise, and a, b are parameters corresponding to the contributions of photon shot noise and thermal noise, respectively. As image brightness decreases, the effect of photon shot noise becomes more significant, while the effect of thermal noise remains relatively small.

Figure S4 presents a comparison of simulated imaging results under different noise parameter settings. When parameter a is fixed and b is doubled, little change is observed in the resulting results. This is because the metalens imaging results are relatively dark, rendering the variation in thermal noise less impactful. In contrast, photon shot noise has a greater influence on the image. As shown in **Fig. S4a, S4c** and **S4e**, increasing a from 0.1 to 0.3 amplifies the noise effect

176 in the simulations. By comparing these simulated imaging results with real captured images, the
177 final noise parameters for dataset generation are determined to be $a = 0.2$, $b = 10$.

Note S5. Gamma Transformation for Low-Light Imaging Simulation.

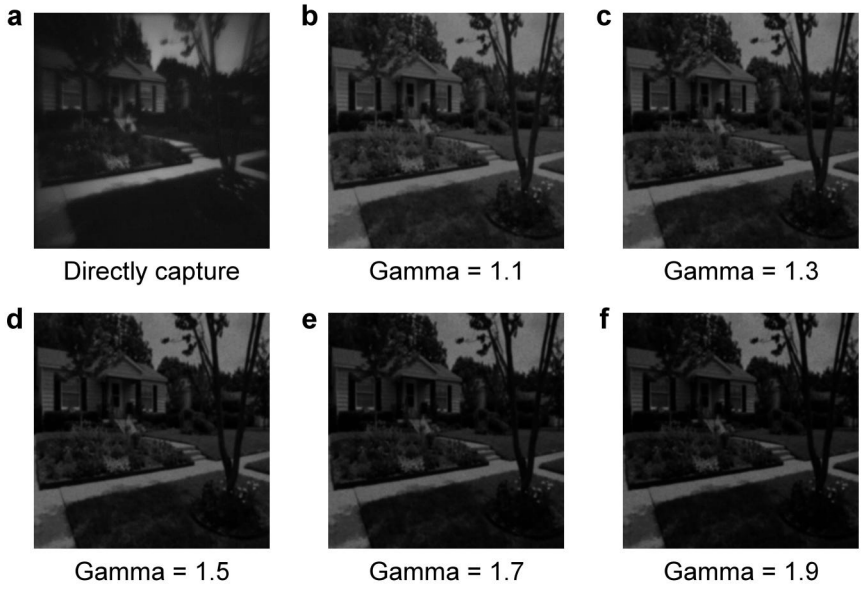

Figure S5 | Effect of different Gamma parameters on the simulated imaging results. a, Metalens imaging captured directly by the CMOS sensor. **b-f,** Simulated imaging results with Gamma parameters set to 1.1, 1.3, 1.5, 1.7, and 1.9, respectively. To better simulate the overall intensity attenuation during metalens imaging, the brightness of all images was reduced by 20%.

After applying PSF convolution and noise augmentation to simulate the imaging system, a nonlinear darkening transformation is required to capture the response characteristics of both the image sensor and the human visual system under low-light conditions, particularly in regions of medium to high brightness. This is achieved by applying a power-law (Gamma) transformation to the normalized pixel intensity. Let the normalized input pixel intensity after the previous processing steps be $\mathbf{I}_{\text{input}}(\mathbf{r})$ (ranging from [0,1]), then the output pixel $\mathbf{I}_{\text{output}}(\mathbf{r})$ is given by:

$$\mathbf{I}_{\text{output}}(\mathbf{r}) = \Gamma\left[\mathbf{I}_{\text{input}}(\mathbf{r})\right] = \left[\mathbf{I}_{\text{input}}(\mathbf{r})\right]^{\gamma} \tag{S4}$$

This transformation serves two purposes in low-light simulation. First, when $\gamma > 1$, it compresses the range of brighter regions while enhances details in the darker areas. Second, the

power-law curve closely approximates the typical response curve of camera sensors, allowing for a more realistic representation of imaging characteristics under low-light conditions. In **Fig. S5**, we present the effects of different Gamma parameters on the final simulated imaging results. It can be observed that when γ is close to 1, the image undergoes only mild darkening, with darker regions remaining clearly visible. However, when γ approaches 2, the overall brightness significantly drops, better approximating extremely low-light conditions, though this may lead to the loss of certain image details. Therefore, we select $\gamma = 1.7$ as the final setting for simulating metalens imaging.

Note S6. Depth-of-field analysis of the PINS system.

The depth of field (DoF) of the PINS system can be estimated using a geometric-optics model based on image-plane shift. According to the Gaussian lens formula, when the object distance u changes, the corresponding image distance v can be expressed as:

$$\frac{1}{f}=\frac{1}{u}+\frac{1}{v} \tag{S5}$$

where f is the focal length of the metalens. When the image distance changes from v to v', the corresponding object distance u' changes to:

$$u^{'}=\frac{f\cdot v^{'}}{v^{'}-f} \tag{S6}$$

An image is considered acceptably sharp if the image plane deviation does not exceed the maximum allowable shift $\Delta v_{\max}$, which is determined by the permissible circle of confusion c and the F-number of the lens: $\Delta v_{\max} = c\times F/\#$. In the present metalens array, each metalens has a diameter of D = 1 mm and a focal length of f = 1.6 mm, yielding a F-number of $F/\# = f/D$ = 1.6. The CMOS sensor has a pixel size of p = 2.2 μm. Considering that the permissible circle of confusion c is typically taken as $2\times p$, $\Delta v_{\max}$ is then set as 7.04 μm, which means that if the image plane deviation does not exceed ±7.04 μm, the image remains acceptably sharp. For the case where the metalens-CMOS distance v is set to f, the corresponding near limit of the DoF can be calculated. The corresponding object distance u' satisfies:

$$u>u_{\text{near}}=\frac{f\cdot v^{'}}{v^{'}-f}=\frac{f\cdot(f+\Delta v_{\max})}{\Delta v_{\max}}=0.37\ \text{m} \tag{S7}$$

Therefore, with the image distance fixed at the focal length, the system maintains acceptable sharpness for object distances from approximately 0.37 m to infinity.

The two adjustment screws are used for fine alignment of the metalens-to-CMOS distance, so that the actual image distance can be tuned as close as possible to the designed focal length. In practice, however, due to fabrication and assembly errors, the actual image distance may still deviate slightly from the ideal focal length. Assuming that the actual metalens-to-CMOS distance is v_0, the corresponding DoF range is determined by the near and far limits associated with the allowable image-plane deviation:

$$u_{\text{near}} = \frac{f \cdot (v_0 + \Delta v_{\max})}{v_0 + \Delta v_{\max} - f}, u_{\text{far}} = \frac{f \cdot (v_0 - \Delta v_{\max})}{v_0 - \Delta v_{\max} - f} \tag{S8}$$

where $u_{\text{far}} = \infty$ when v_0-$\Delta v_{\max} \leq f$. **Table S1** summarizes the resulting DoF ranges for several representative values of v_0 relative to the nominal focal length f = 1.6 mm.

Table S1 | Depth of field ranges for different image distances.

Image distance v_0	Near limit object distance u_{near} (m)	Far limit object distance u_{far} (m)
f	0.37	∞
f + 10 μm	0.15	0.87
f + 15 μm	0.12	0.32
f + 20 μm	0.10	0.20
f + 25 μm	0.08	0.14

These results show that the depth of field is maximized when the image distance is set as close as possible to the focal length. As the image plane shifts away from the focal plane, the in-focus object-distance range rapidly narrows. The screw mechanism provides continuous depth adjustment for focus alignment, with a resolution of approximately 1.1 μm per 1° of rotation, offering sufficient positioning accuracy for practical focus tuning within the acceptable tolerance range.

Note S7. Comparison of distortion-corrected experimentally captured image and simulated image using the PSF model.

First, we quantified the similarity between the corresponding image pairs in **Fig. 2e** by calculating the structural similarity index measure (SSIM), mean absolute error (MAE), and mean squared error (MSE), as summarized in **Table S2**.

Table S2 | Quantitative similarity metrics for the corresponding image pairs shown in Fig. 2e.

	SSIM	MAE	MSE
Image 1	0.741802	0.047315	0.007987
Image 2	0.791876	0.044247	0.008249
Image 3	0.803224	0.031747	0.005888

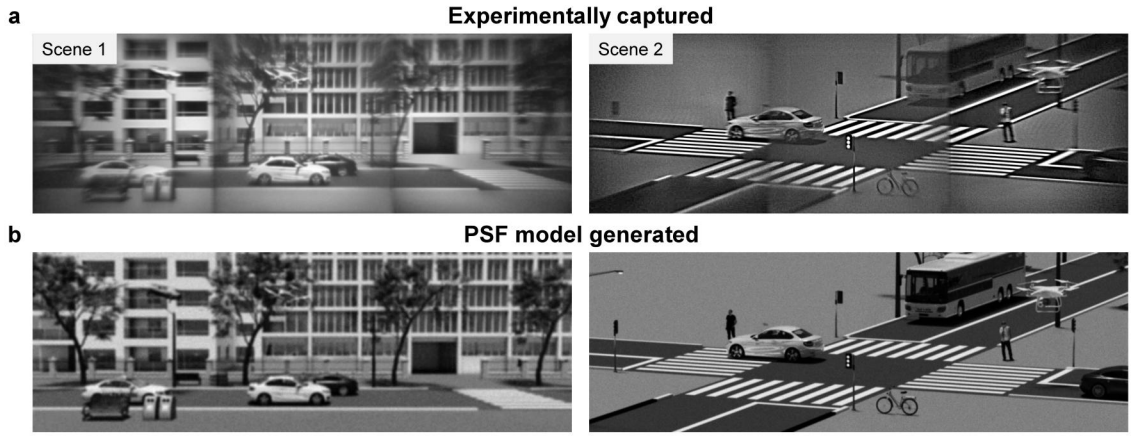

Figure S6 | Comparison of distortion-corrected experimentally captured 135°-FoV image and simulated 135°-FoV image using PSF model. a, Experimental captured image by the metalens array after reconstruction. **b,** Simulated image of the same scene generated using only the central 45° PSF.

To further verify the validity of the PSF-based convolution imaging model for wide-FoV motion sensing, we compared distortion-corrected experimentally captured wide-FoV images with simulated wide-FoV images generated using only the experimentally measured central PSF. The experimental wide-FoV images were reconstructed by sub-image extraction, distortion correction, and image stitching. The simulated wide-FoV images were generated using the PSF captured by the central-FoV metalens under normal illumination for the same scenes.

As shown in **Fig. S6**, the simulated images agree well with the reconstructed experimental images in overall scene structure, local blur, brightness distribution, and object contours. To further quantify the similarity, we calculated the SSIM, MAE, and MSE for two representative scenes, as summarized in **Table S3**. The results indicate that, although some residual differences remain due to experimental noise, illumination variation, and reconstruction error, the central-PSF-based model can still reproduce the main image characteristics of the wide-FoV system with reasonable accuracy. These results support the use of the central PSF as a practical approximation for training-data generation in the present wide-FoV motion-sensing framework.

Table S3 | Quantitative similarity metrics for the corresponding image pairs shown in Fig. S6.

	SSIM	MAE	MSE
Scene 1	0.6476	0.1394	0.0387
Scene 2	0.7164	0.0913	0.0172

Note S8. Analysis of the effects of imaging degradation on optical-flow estimation.

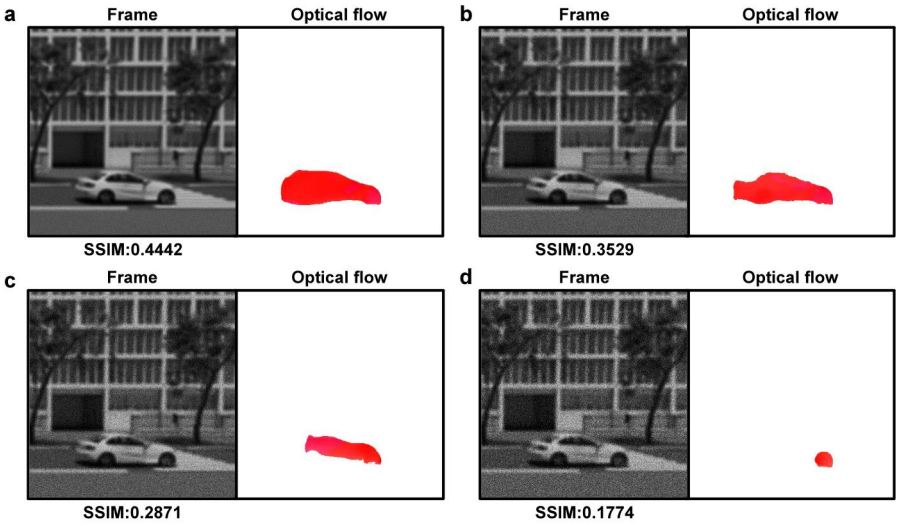

Figure S7 | Effect of imaging noise on optical-flow estimation. Representative optical-flow results under different imaging noise levels, with SSIM values of (**a**) 0.4442, (**b**) 0.3529, (**c**) 0.2871 and (**d**) 0.1774.

To evaluate the effect of imaging noise on optical-flow estimation, we introduced different levels of noise into the input images and compared the corresponding motion-estimation results, as shown in **Fig. S7**. The image quality is quantified by the structural similarity index (SSIM), which decreases from 0.4442 in **Fig. S7a** to 0.1774 in **Fig. S7d** as the noise level increases. It can be seen that increasing noise gradually degrades the completeness and accuracy of the estimated motion field. When the noise becomes severe, the detected motion region shrinks significantly and partial motion information is lost. These results indicate that imaging noise directly affects the reliability of optical-flow estimation and thus can degrade subsequent motion analysis performance.

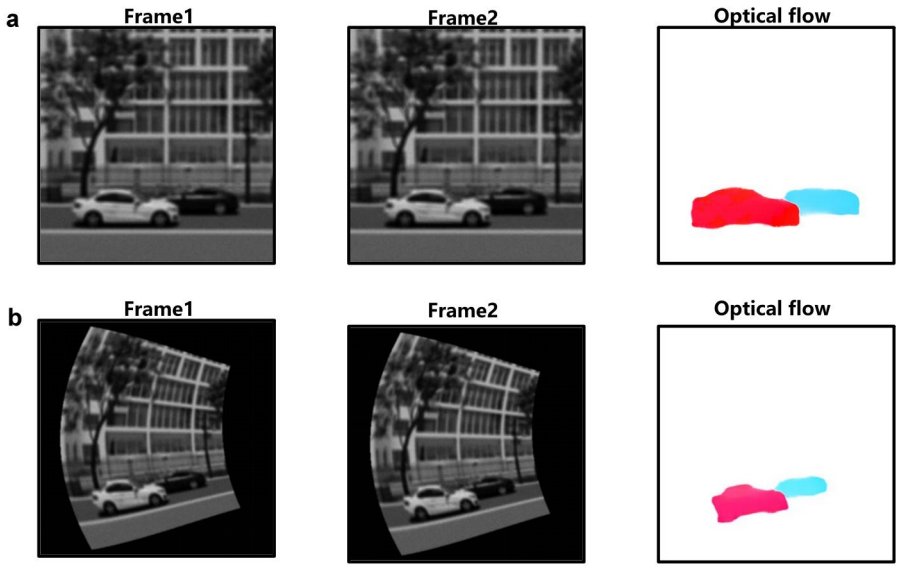

Figure S8 | Effect of image distortion on optical-flow estimation. Comparison of optical-flow results for (**a**) undistorted and (**b**) distorted image pairs.

To evaluate the influence of image distortion on optical-flow estimation, we compared the motion-estimation results obtained from undistorted and distorted image pairs, as shown in **Fig. S8**. In **Fig. S8a**, the input images are free of geometric distortion, and the corresponding optical-flow result accurately preserves the shape and relative position of the moving objects. In **Fig. S8b**, geometric distortion is introduced into the input image pair, leading to visible deformation of the scene structure. As a result, the estimated optical-flow vectors exhibit errors in both direction and magnitude. Although subsequent geometric rectification can correct the spatial position of the motion field, it cannot fully recover the distortion-induced errors in the optical-flow direction and magnitude. This result indicates that image distortion not only affects image geometry, but also degrades the physical accuracy of optical-flow estimation, further justifying the distortion-correction step in our wide-angle image reconstruction pipeline before motion analysis.

Note S9. Structural details of MMS deep neural network.

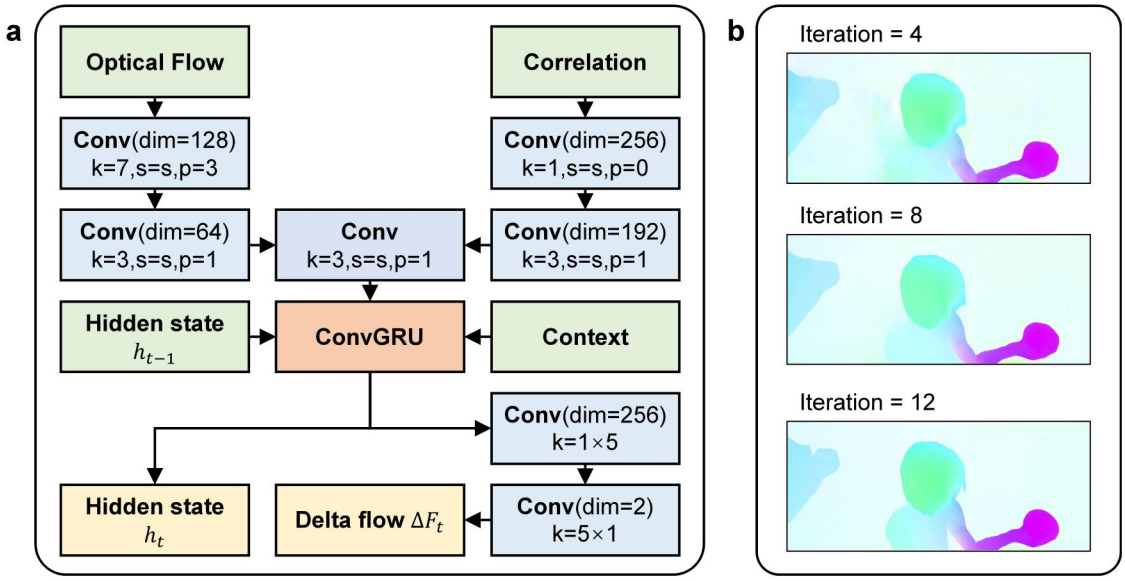

Figure S9 | Iterative update block of Meta-Motion Sense (MMS) neural network. a, Schematic of the process flow and structure for a single optical flow update. The input to the module comprises four components: the current optical flow estimate, correlation features, the hidden state from the previous iteration, and context features. The module outputs the updated hidden state and the incremental optical flow correction. **b,** Impact of the number of iterations on the accuracy of optical flow estimation. The input data is generated using PSF-based simulated imaging, with specific comparison values shown in **Table S4.**

Table S4 | Accuracy of optical flow estimation under different iterations.

	EPE	APE	1-pixel	3-pixel	5-pixel
Iteration=4	5.86	6.53	0.87	0.96	0.98
Iteration=8	4.11	4.69	0.89	0.98	0.99
Iteration=12	3.22	2.69	0.92	0.99	0.99
Iteration=16	3.22	2.69	0.92	0.99	0.99

Note: EPE (End Point Error): the Euclidean distance between the predicted and ground truth flow vectors at each pixel. APE (Angular Projection Error): the angular error of the flow vectors, measuring the deviation between the predicted and ground truth flow directions. The 1-pixel, 3-pixel, and 5-pixel metrics indicate the

298 percentage of estimated flow vectors whose end-point error is within 1, 3, or 5 pixels, respectively. These
299 metrics help evaluate the performance of the motion estimation method at varying levels of accuracy.

300 From the beginning, the initial optical flow map is set to be zero at every pixel. At t-th iteration,
301 the estimated optical flow F_k is used to predict a new sampling position $\mathbf{x}' = \mathbf{x}+F_k(\mathbf{x})$,
302 representing the potential correspondence in the second frame I_2 for a given pixel position $\mathbf{x}$ in
303 the first frame I_1. Since the estimated optical flow value are typically sub-pixel and the
304 corresponding positions are not aligned with integer pixel coordinates, direct indexing into the
305 correlation volume is infeasible. As a result, the interpolation is required at the sampling
306 positions. Around each sampling position, a local neighbourhood with radius r is extracted from
307 every level of the 4D correlation pyramid using bilinear interpolation, yielding multi-scale
308 matching responses. These responses are concatenated with the present estimated flow features,
309 along with the contextual features extracted from the first frame I_1 via the Dual-Stage Multi-
310 Scale Composite Encoder (DSMSCE) module.

311 The resulting feature combination is subsequently fed into an update block composed of a
312 Convolutional Gated Recurrent Unit (ConvGRU) with shared weights, as detailed in **Fig. S9a**.
313 This module is designed to iteratively refine the optical flow estimation over multiple update
314 steps. By replacing the fully connected layers in a standard GRU with convolutional operations,
315 the ConvGRU preserves the spatial structure of the input feature maps. During each iteration, the
316 ConvGRU updates the hidden state and predicts an incremental flow correction, which is added
317 to the previous estimate to refine the optical flow. In each iteration, the network outputs an
318 incremental update to the flow field based on the current flow estimate, correlation features, and
319 context features, gradually approaching the final precise optical flow field. Specifically, in a
320 single iteration update, the network first accepts the current estimated flow and the correlation

features obtained through search. These features are then further processed and concatenated through convolutions with different dimensions. Subsequently, the concatenated features, along with context features and the hidden state from the previous iteration, are input into a ConvGRU module. Taking the *t*-th iteration as an example, the specific formula of the GRU module is:

$$z_t = \sigma\left(\mathrm{Conv}_{3\times3}\left(\left[h_{t-1}, x_t\right], W_z\right)\right) \tag{S9}$$

$$r_t = \sigma\left(\mathrm{Conv}_{3\times3}\left(\left[h_{t-1}, x_t\right], W_r\right)\right) \tag{S10}$$

$$\tilde{h}_t = \tanh\left(\mathrm{Conv}_{3\times3}\left(\left[r_t \quad h_{t-1}, x_t\right], W_h\right)\right) \tag{S11}$$

$$h_t = \left(1-z_t\right) \quad h_{t-1} + z_t \quad \tilde{h}_t \tag{S12}$$

where x_t represents the concatenated input features, $h_{t\text{-}1}$ is the hidden state from the previous time step, σ denotes the sigmoid activation, tanh is the hyperbolic tangent function, and $\odot$ represents element-wise multiplication. The initial hidden state and context features are obtained from the DSMSCE based on the first frame. The hidden state output by the GRU module is then passed through two convolutional layers to predict the flow increment ΔF_t for the current iteration. The new optical flow estimate is then calculated as $F_{t+1} = F_t + \Delta F_t$. Moreover, since the output flow field resolution is 1/8 of the input image resolution, we employ a learnable convolutional weighted interpolation method to recover the original resolution. During up-sampling, the network first predicts a learnable mask of size 8×8×9, and by applying SoftMax weighting to the 9 neighbouring pixels, the low-resolution flow is up-sampled to full resolution. By repeating this iterative process, the optical flow estimation is progressively improved.

Additionally, we studied the impact of the number of iterations on the optical flow estimation accuracy. We selected two clean images from the Sintel dataset and generated simulation results close to metalens imaging using the PSF-based simulation method. As shown in **Fig. S9b**, as the

number of iterations increases from 4 to 12, the optical flow estimation results improve, and the details gradually become clearer. The specific metrics in **Table S4** further confirm this trend. However, when the number of iterations exceeds 12, the improvement in optical flow estimation accuracy becomes negligible. Therefore, we set the number of iterations to 12 for metalens optical flow estimation.

Note S10. Structural details of DSMSCE module.

Given an input feature map $X_1 \in {}^{H_1 \times W_1 \times C}$, the outputs of the parallel pooling branches can be expressed as:

$$P_m = \text{Upsample}\left[\text{AvgPool}(X_1; \text{size} = m \times m)\right], m \in \{1,2,3\} \quad \text{(S13)}$$

The identity mapping branch is denoted as $P_0 = X_1$. By concatenating this branch with the outputs of the parallel pooling paths, a combined feature representation $Z_1 = \text{Concat}(P_0, P_1, P_2, P_3) \in {}^{H_1 \times W_1 \times 4C}$ is formed. A convolution operation is then applied to this concatenated feature map to reduce the dimensionality, yielding in the initial fused representation.

The DASPP module consists of a 1×1 standard convolution, multiple 3×3 Atrous convolutions with different dilation rates, and a global average pooling branch. Atrous convolution is a method to expand the receptive field by introducing gaps (or dilation) between the sampling points of a standard convolution operation. Given the intermediate feature input $X_2 \in {}^{H_2 \times W_2 \times C'}$, the output of the 3×3 Atrous convolution branch can be expressed as:

$$A_r(i,j) = \sum_{m,n=-1}^{1} W_r(m,n) X_2\left[i + rm, j + rn\right], r \in \{6,12,18\}, W_r \in {}^{3 \times 3 \times C'} \quad \text{(S10)}$$

where $A_r(i,j)$ denotes the value at position (i,j) of the output feature map obtained by applying an Atrous convolution with dilation rate r. The convolutional kernel weight at position (m,n) for dilation rate r is denoted as $W_r(m,n)$, where $m,n \in \{-1,0,1\}$, covering all positions within the 3×3 kernel. The parameter r defines the dilation rate, which controls the sampling interval, thus achieving multi-scale receptive fields. Each kernel W_r has a shape of $3 \times 3 \times C'$, indicating that each output channel is associated with a distinct 3×3 filter. In our implementation, the dilation rates are set to r = 6,12,18, to obtain the outputs from the dilated convolution branches, denoted as A_1, A_2, A_3, respectively. Let A_0 represent the output of the standard 1×1 convolution branch,

364 and A_4 denote the output of the global average pooling branch followed by a 1×1 convolution.

365 The final output of the DASPP module is obtained by concatenating all branches:

$$Y = \mathrm{Conv}_{1\times1}\left(\mathrm{Concat}\left(A_0, A_1, A_2, A_3, A_4\right)\right), Y \in \quad ^{H_2\times W_2\times C'} \tag{S11}$$

366

Note S11. Sequence-wise exponentially weighted L_1 loss for optical flow estimation.

During training, our network utilizes a sequence-wise L_1 loss function to supervise the optical flow estimation process. Specifically, the network generates a sequence of N optical flow predictions through iterative refinement, where the prediction at the i-th iteration is denoted as f_i, and the ground-truth flow is denoted as f_{gt}. The overall loss function is formulated as follows:

$$L = \sum_{i=1}^{N} \rho^{i-N} \left\| f_{\text{gt}} - f_i \right\|_1 \tag{S16}$$

where $\| \|_1$ denotes the pixel-wise L_1 norm, which measures the absolute error between the predicted and ground-truth flow fields for every pixel. The parameter N represents the total number of iterations. The parameter ρ is an exponential decay factor, ranging from 0 to 1 (set to 0.8 here). This weighting mechanism assigns greater importance to later iterations closer to the final output, while earlier predictions receive lower weights. By supervising all intermediate predictions with exponentially increasing weights, this loss function accelerates network convergence and improves the quality of flow estimation at each step. Furthermore, this sequence-wise supervision enhances the robustness and generalization ability of the model, ensuring reliable optical flow prediction even with fewer refinement steps.

Note S12. Characterization of the motion detection limits in PINS.

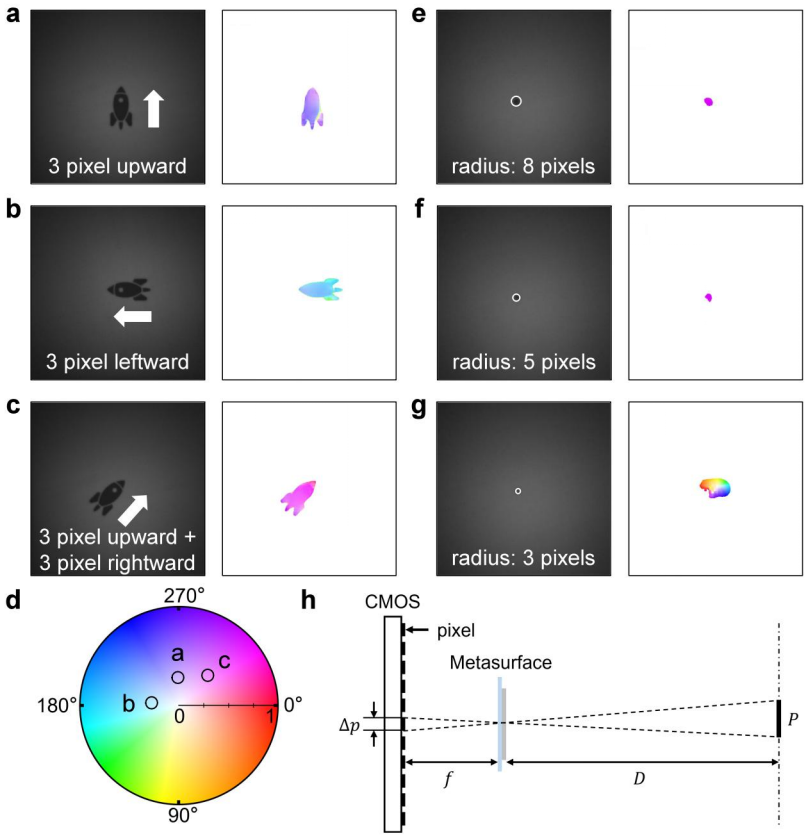

Figure S10 | Characterization of the extreme motion detection capabilities of the Meta-Motion Sense (MMS) deep neural network for subtle displacements and small targets. Detected optical flow results for subtle displacements of the rocket pattern moving towards three different directions: **a,** 3 pixels upward, **b,** 3 pixels leftward, and **c,** sequentially 3 pixels and 3 pixels rightward. **d,** Three movements shown in (**a**) to (**c**)are marked on the colour wheel. Detected optical flow results for small circular targets with radii of **e,** 8 pixels, **f,** 5 pixels and **g,** 3 pixels, respectively, moving 15 pixels to the right. All the captured images are 701 × 701 pixels in size. **h,** Schematic illustration for spatial resolution estimation. Δp is the image size on CMOS sensor, f is the focal length of metalens, D is the object distance, and P is the corresponding object size.

To further investigate the motion detection limits of PINS, we designed test scenarios featuring minute displacements and tiny size moving targets, and analysed the resulting optical flow outputs. To assess PINS's sensitivity to minimal motion, we employed a rocket pattern, and

tested its displacement in three directions: vertical, horizontal and diagonal. **Figures S10a** and **S10b** show two sequential frames and their corresponding optical flow results when the target is displaced by 3 pixels upward and leftward, respectively. **Figure S10c** illustrates the results for a diagonal displacement, in which the object moves 3 pixels upward and 3 pixels rightward, resulting in an effective shift of approximate 4.2 pixels. These results confirm that PINS is capable of accurately reconstructing the object's shape and reliably capturing both the direction and magnitude of motion, even at the level of a single pixel moving on CMOS. The motion information corresponding to the optical flow results in **Fig. S10a to S10c** is indicated in the colour wheel shown in **Fig. S10d**.

Furthermore, to evaluate the detection limit for small-scale moving objects, we translated small targets with radii of 8, 5, and 3 pixels, respectively, by 15 pixels to the right and analysed the corresponding optical flow results. As shown in **Fig. S10e**, for the 8-pixel-radius target, the MMS neural network accurately detected both the motion and its direction, indicating reliable performance. When the radius was reduced to 5 pixels, minor distortions emerged around the target's periphery, however, the motion remained clearly identifiable, as shown in **Fig. S10f**. In contrast, for the smallest target with a radius of 3 pixels, the network failed to robustly capture the motion, and the resulting optical flow exhibited considerable artifacts and noise, as illustrated in **Fig. S10g**. These findings indicate that PINS can effectively detect moving objects with image sizes as small as 5 pixels in diameter on CMOS.

The aforementioned displacements and object sizes are quantified in terms of pixel units on the CMOS sensor but can be readily converted into real-world spatial dimensions. This conversion depends on both the pixel size (2.2 μm) of the CMOS sensor and the focal length (1.6 mm) of the integrated metalens, as illustrated in **Fig. S10h**. After conversion, each pixel on the

CMOS sensor corresponds to an angular resolution of approximately 1.2 mrad (~0.07°). Accordingly, the minimum detectable displacement of the PINS system is about 4 cm at a distance of 10 meters. Furthermore, the smallest object size that can be reliably identified by PINS corresponds to approximately 10 cm at the same distance.

Note S13. Detailed formulation of the Motion Trajectory Prediction (MTP) method.

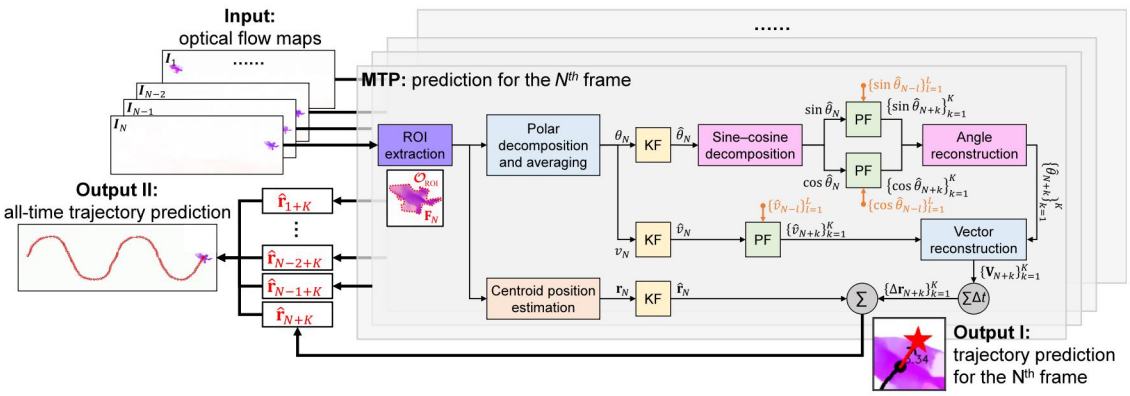

Figure S11 | Detailed processing flowchart of single-object trajectory prediction using MTP framework. The MTP framework consists of four key stages: region of interest (ROI) extraction, current centroid position estimation, velocity pre-processing and curve fitting, and future centroid position estimation with trajectory extrapolation. KF: Kalman filtering. PF: polynomial fitting.

Figure S11 illustrates the process of predicting the future trajectory of a single object based on the preceding sequence of optical flow maps $\{I_1, I_2, \ldots, I_N\}$, with I_N denoting the current frame. The process begins with ROI extraction, where the dominant moving region in each optical flow map is identified by thresholding the flow magnitude followed by connected-component analysis, as illustrated by the red dashed outline in **Fig. S11**. The largest connected region 0_{ROI} is selected as the target, effectively suppressing background motion and noise. Next, from the extracted region 0_{ROI}, the current object's centroid position $\mathbf{r}_N$ is estimated as the spatial average of pixel coordinates:

$$\mathbf{r}_N = (x_N, y_N) = \frac{1}{M}\left(\sum_{i \in 0_{\mathrm{ROI}}} x_i, \sum_{i \in 0_{\mathrm{ROI}}} y_i \right) \tag{S17}$$

where M denotes the total number of pixels contained within the region 0_{ROI}. Subsequently, velocity information is extracted from the optical flow map in the region of interest,

preprocessed, and fitted using polynomial regression. Given the optical flow map $\mathbf{F}_N$ within 0_{ROI}, a polar decomposition and averaging are performed to obtain the velocity magnitude v_N and direction angle θ_N of the target object:

$$v_N = \frac{1}{M} \sum_{f \in \mathbf{F}_N} \sqrt{f_x^2 + f_y^2} \tag{S18}$$

$$\theta_N = \frac{1}{M} \sum_{f \in \mathbf{F}_N} \operatorname{atan2}\left(f_y, f_x\right) \tag{S19}$$

where atan2(·) denotes the arctangent function with two arguments, which takes into account the sign of both the numerator and denominator to determine the correct quadrant of the angle. To suppress frame-to-frame noise, the current centroid position $\mathbf{r}_N$, velocity magnitude v_N and angle θ_N are then independently smoothed using Kalman filters, resulting in the filtered estimates $\hat{\mathbf{r}}_N$, $\hat{v}_N$, and $\hat{\theta}_N$, respectively. The next step involves polynomial fitting of the filtered velocity magnitude $\hat{v}_N$ and angle $\hat{\theta}_N$. Specifically, the filtered velocity magnitude $\{\hat{v}_{N-l}\}_{l=1}^{L}$ and angle $\{\hat{\theta}_{N-l}\}_{l=1}^{L}$ over a recent window of L, from I_{N-L} to I_{N-1}, are retained and fitted using weighted polynomial regression to capture the temporal trend of motion. To emphasize recent motion and reduce the risk of overfitting to outdated motion patterns, a linearly increasing temporal weighting scheme is adopted, assigning greater weights to frames closer to the current time index. The sequence of velocity magnitudes $\{\hat{v}_{N-l}\}_{l=1}^{L}$ is directly modeled as a quadratic function of time index n:

$$v = a_1 n^2 + b_1 n + c_1, v \in \{\hat{v}_{N-l}\}_{l=1}^{L} \tag{S20}$$

where a_1, b_1, c_1 are the polynomial coefficients estimated from the weighted least squares fitting. However, when fitting the direction angle $\{\hat{\theta}_{N-l}\}_{l=1}^{L}$, the periodic nature of angular data must be

properly addressed to avoid discontinuities near the 0 or 2π boundary. Accordingly, the velocity angle is first decomposed into its sine and cosine components:

$$\begin{cases} \sin\theta = a_2 n^2 + b_2 n + c_2 \\ \cos\theta = a_3 n^2 + b_3 n + c_3 \end{cases}, \theta \in \{\hat{\theta}_{N-l}\}_{l=1}^{L} \tag{S21}$$

These components are then individually fitted using polynomial regression and subsequently recombined via inverse trigonometric reconstruction. This approach ensures angular continuity and improves the numerical stability of both the fitting and prediction processes for directional motion. The fitted curves described in **Eq. (S20)** and **Eq. (S21)** are employed to extrapolate the velocity magnitude $\{\hat{v}_{N+k}\}_{k=1}^{K}$ and angle $\{\hat{\theta}_{N+k}\}_{k=1}^{K}$ from frame N+1 to frame N+K:

$$\begin{cases} v_{N+n+1-L} = a_1 n^2 + b_1 n + c_1 \\ \sin\theta_{N+n+1-L} = a_2 n^2 + b_2 n + c_2 \\ \cos\theta_{N+n+1-L} = a_3 n^2 + b_3 n + c_3 \end{cases}, n \in [L, L+K-1] \tag{S22}$$

From these extrapolated quantities, the future velocity vectors $\{\mathbf{V}_{N+k}\}_{k=1}^{K}$ can be reconstructed by converting the polar representation $\{(\hat{v}_{N+k}, \hat{\theta}_{N+k})\}_{k=1}^{K}$. Furthermore, the predicted object's position $\hat{\mathbf{r}}_{N+k}$ at frame I_{N+K} is obtained by numerically integrating the predicted velocity vectors over fine-grained temporal substep $\Delta t = T_{\text{frame}}/5$, spanning the interval [N+1, N+k]:

$$\hat{\mathbf{r}}_{N+k} = \hat{\mathbf{r}}_N + \Delta\mathbf{r}_{N+k} = \hat{\mathbf{r}}_N + \sum_{m=1}^{5k} \mathbf{V}_{N+m/5} \Delta t \tag{S23}$$

Here, although the intermediate substep index N+m/5 is not an integer, it can still be directly substituted into **Eq. (S22)** to evaluate the extrapolated velocity using the fitted polynomial curves. This substep-based integration scheme mitigates accumulation errors and improves the numerical stability compared to direct vector summation over full frames. Finally, the predicted positions and motion trajectory of the target from frame N+1 to frame N+K can be obtained, as illustrated

in Output I of **Fig. S11**. In the figure, the five-pointed star denotes the predicted position at frame $N+K$; the red solid line represents the predicted trajectory over the interval $N+1$ to $N+K$; the black filled circle indicates the centroid position at the current frame; the black solid line traces the ground-truth trajectory of the centroid; and the black arrow and numeric label indicate the current velocity direction and magnitude (in units of pixels per second). Furthermore, by recording the predicted position at the furthest forecasted frame for each input frame, we can construct an all-time prediction trajectory, which enables direct comparison with the ground-truth motion path, as illustrated in Output II of **Fig. S11**.

Note S14. Analysis of the influence of fitting parameters on trajectory prediction performance.

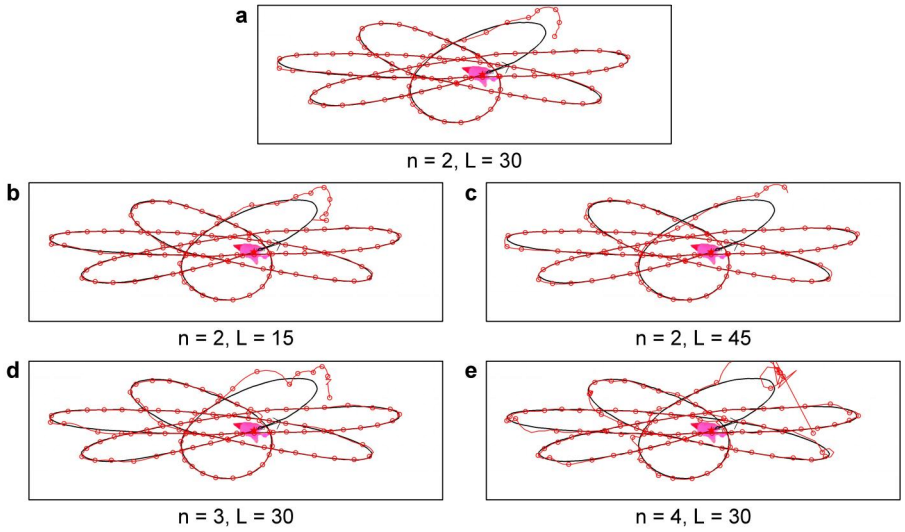

Figure S12 | Influence of the polynomial fitting order and recent window length on trajectory prediction performance. a, Electron trajectory prediction results for 15 frames ahead when the polynomial fitting order is $n = 2$ and the length of the recent window is $L = 30$, consistent with Fig. 5d. **b-c,** Electron trajectory prediction results with fixed polynomial order $n = 2$ and window length set to $L = 15$ and $L = 45$. **d-e,** Electron trajectory prediction results with fixed window length $L = 30$ and polynomial order set to $n = 3$ and $n = 4$.

To evaluate the sensitivity of the trajectory prediction performance to the fitting parameters, we conducted a series of controlled experiments by varying the polynomial fitting order n and the length of the recent window L. **Figure S12** summarizes the prediction results under different parameter configurations. Among all tested configurations, the setting shown in **Fig. S12a** with polynomial order $n = 2$ and recent window length $L = 30$ yields the best overall performance. It achieves a good balance between smoothness and responsiveness, and is consistent with the prediction result presented in **Fig. 5d** of the main text.

To investigate the influence of the recent window length L, we fixed the polynomial order at $n = 2$ and compared prediction results under different L values. In contrast, **Fig. S12b** shows the result when $L = 15$ with $n = 2$ fixed. Due to the limited number of historical data available for fitting, the prediction becomes more susceptible to local noise, leading to noticeable fluctuations and irregular jitter in the predicted trajectory. This indicates underfitting caused by insufficient context for the polynomial model. When the window length is extended to $L = 45$ (**Fig. S12c**), the prediction starts to exhibit overfitting. Although more data points are included in the fitting process, the model becomes less responsive to recent directional changes, resulting in larger deviations at trajectory turning points. This illustrates the trade-off between temporal resolution and stability.

Furthermore, we analysed the impact of increasing the polynomial fitting order while keeping $L = 30$ fixed. As shown in **Fig. S12d and S12e**, when the order is raised to $n = 3$ and $n = 4$, the predicted trajectories become increasingly erratic and less physically plausible. This degradation is likely due to Runge's phenomenon, where higher-order polynomials tend to exhibit large oscillations, especially near the boundaries of the fitting interval. In the presence of noise or subtle fluctuations in the input data, high-order models tend to amplify such variations, leading to unstable and unreliable predictions.

Note S15. Quantitative RMSE evaluation of trajectory prediction accuracy.

Table S5 summarizes the quantitative RMSE evaluation of the trajectory prediction accuracy for the single-object cases in **Fig. 5** and the multi-object case in **Fig. 6**. For the single-object cases in **Fig. 5**, the reference position is defined as the centroid position extracted at the corresponding future frame. For the multi-object case in **Fig. 6**, the RMSE values are reported separately for each tracked object ID. These results demonstrate that the proposed MTP framework achieves accurate short-term trajectory prediction for both single-object and multi-object cases. Meanwhile, the prediction error increases for more complex trajectories and longer forecasting horizons, as expected, due to accumulated extrapolation uncertainty.

Table S5 | Quantitative RMSE evaluation of the trajectory prediction accuracy in Fig. 5 and Fig. 6.

	$RMSE_X$ (pixels)	$RMSE_Y$ (pixels)	$RMSE_{2D}$ (pixels)
1 frame ahead in Fig. 5b	1.51	1.09	1.86
15 frames ahead in Fig. 5b	20.54	14.46	25.12
30 frames ahead in Fig. 5b	41.82	28.08	50.37
1 frame ahead in Fig. 5c	5.02	3.16	5.93
15 frames ahead in Fig. 5c	71.68	45.36	84.83
30 frames ahead in Fig. 5c	137.68	94.35	166.90
1 frame ahead in Fig. 5d	10.69	3.17	11.15
15 frames ahead in Fig. 5d	153.47	46.07	160.23
30 frames ahead in Fig. 5d	300.52	88.35	313.24
ID 0 in Fig. 6	10.96	3.83	11.61
ID 1 in Fig. 6	4.02	1.39	4.25
ID 2 in Fig. 6	9.91	2.46	10.21
ID 3 in Fig. 6	38.32	5.29	38.68
ID 4 in Fig. 6	7.47	2.29	7.81

Note: "$RMSE_X$" denotes the root mean square errors of the predicted positions along the *x* direction, "$RMSE_Y$" denotes the root mean square errors of the predicted positions along the *y* direction, "$RMSE_{2D}$" denotes the two-dimensional root mean square error, calculated from the Euclidean distance between the predicted future position and the corresponding reference position at the target frame.

Note S16. Flexible trade-off between motion-information redundancy and spatial coverage by tilt-phase design.

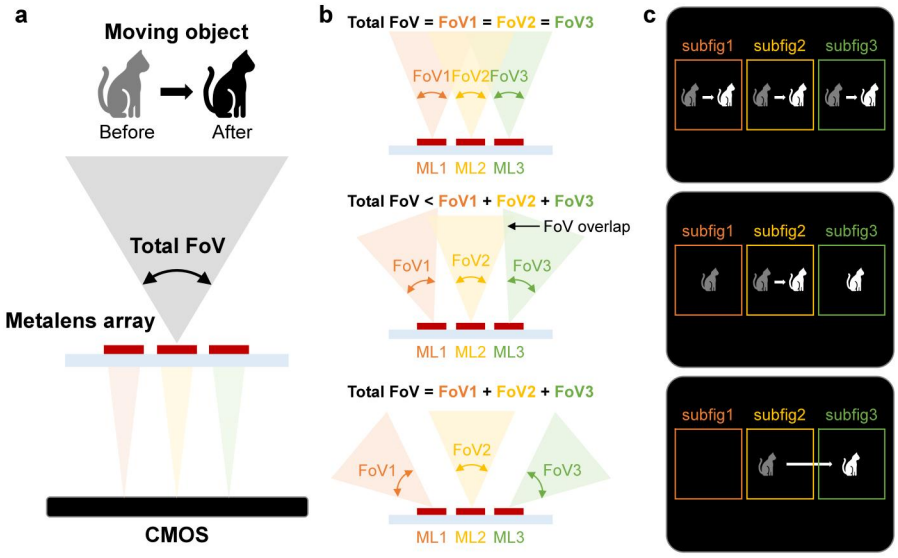

Figure S13 | View-angle engineering of the metalens array by tilt-phase design. a, Schematic illustration of a dynamic scene observed by the metalens array. **b,** Different view-angle allocation schemes realized by assigning different linear tilt phases to the three metalenses. **c,** Corresponding imaging results under the different phase-design schemes.

Figure S13 illustrates the view-angle engineering enabled by adding different tilt phases to the metalenses in the array. By adjusting the tilt phase assigned to each metalens, the angular sectors corresponding to different sub-images can be redistributed, leading to different imaging modes. When the view angles of adjacent metalenses are more concentrated and partially overlap, the system captures the same moving target in multiple sub-images, thereby providing richer motion-related information but a smaller overall field of view. In contrast, when the view angles are more separated and the overlap between adjacent sub-images is reduced, the total field of view is enlarged and more spatial information can be covered. Therefore, this design provides a flexible trade-off between motion-information redundancy and wide-angle spatial coverage.

Note S17. Extension to stereo-vision depth measurement based on metalenses.

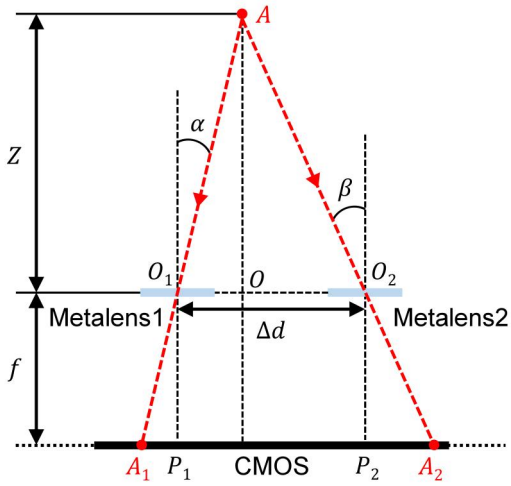

Figure S14 | Principle of stereo-vision depth measurement based on metalenses.

A single-lens imaging system can only provide two-dimensional (2D) projection information of a three-dimensional (3D) scene. Since depth information is inherently lost in the projection process, it is difficult to estimate the absolute distance of objects using a monocular image alone without additional assumptions or constraints. In contrast, a stereo-vision system mimics human binocular perception by using two spatially separated metalenses to simultaneously capture the same scene from different viewpoints, enabling the extraction of depth information from a 3D scene.

Figure S14 shows a stereo-vision system based on two metalenses. The two optical centres O_1 and O_2 of metalens1 and metalens2 are aligned along a common baseline and spaced by a distance $\Delta d = |O_1O_2|$. The imaging point A_1 is formed by metalens1 from the object point A with an incident angle α. The position of A_1 can be calculated from the projection function $A_1P_1 = P_f(\alpha)$. The projection function is dependent on the phase design of the metalens—for instance, $P_f(\alpha) = -f\tan\alpha$ for a hyperbolic phase profile, and $P_f(\alpha) = -f\sin\alpha$ for a quadratic phase profile.

Accordingly, the incident angle α can be reconstructed by inverting the projection function, yielding:

$$\alpha = P_f^{-1}\left[\left|A_1P_1\right|\right] = \begin{cases} \arctan\left[-\dfrac{\left|A_1P_1\right|}{f}\right], & \text{for hyperbolic phase profile} \\ \arcsin\left[-\dfrac{\left|A_1P_1\right|}{f}\right], & \text{for quadratic phase profile} \end{cases}. \tag{S24}$$

Here, $|A_1P_1|$ denotes the image-plane distance between the optical axis and the projection point A_1, and f is the focal length of the metalens. Subsequently, from the triangle $\triangle O_1AO$, it follows that the horizontal distance from O_1 to O is given by $|O_1O| = Z\tan\alpha$, where Z denotes the depth of point A relative to the baseline. Similarly, for metalens2, the corresponding horizontal distance can be expressed as $|O_2O| = Z\tan\beta$, where β is the incident angle of A at metalens2. Finally, based on the geometric relationship $\Delta d = \left|O_1O_2\right| = \left|O_1O\right| + \left|O_2O\right|$ the object depth Z of point A can be computed as:

$$Z = \frac{\Delta d}{\tan\left(\alpha\right) + \tan\left(\beta\right)} = \frac{\Delta d}{\tan\left(P_f^{-1}\left[\left|A_1P_1\right|\right]\right) + \tan\left(P_f^{-1}\left[\left|A_2P_2\right|\right]\right)}. \tag{S25}$$

Movie S1. Single-object trajectory prediction for representative motion patterns.

Movie S1 provides the complete results of the single-object trajectory prediction for three representative motion patterns: sinusoidal, planar spiral, and random trajectories.

Movie S2. Multi-object tracking and trajectory forecasting in complex wide-angle scenes.

Movie S2 presents the complete results of multi-object tracking and trajectory prediction in complex wide-angle scenes. The system robustly segments and identifies multiple independently moving targets, assigns consistent object IDs across frames, and accurately predicts their future trajectories.