M^{3}Cam: Lightweight Super-Resolution via Multi-Modal Optical Flow for Mobile Cameras

Yu Lu*[‡]
Shanghai Jiao Tong University
yulu01@sjtu.edu.cn

Dian Ding*[‡]
Shanghai Jiao Tong University
dingdian94@sjtu.edu.cn

Hao Pan[†]
Microsoft Research Asia
panhao@microsoft.com

Yongjian Fu
Central South University
fuyongjian@csu.edu.cn

Liyun Zhang[‡]
Shanghai Jiao Tong University
zhang_ly@sjtu.edu.cn

Feitong Tan
Simon Fraser University
feitongt@sfu.ca

Ran Wang[‡]
Shanghai Jiao Tong University
wang_r@sjtu.edu.cn

Yi-Chao Chen[‡]
Shanghai Jiao Tong University
yichao@sjtu.edu.cn

Guangtao Xue[‡]
Shanghai Jiao Tong University
gt_xue@sjtu.edu.cn

Ju Ren
Tsinghua University
renju@tsinghua.edu.cn

Abstract

The demand for ultra-high-resolution imaging in mobile phone photography is continuously increasing. However, the image resolution of mobile devices is typically constrained by the size of the CMOS sensor. Although deep learning-based super-resolution (SR) techniques have the potential to overcome this limitation, existing SR neural network models require large computational resources, making them unsuitable for real-time SR imaging on current mobile devices. Additionally, cloud-based SR systems pose privacy leakage risks. In this paper, we propose M^{3}Cam, an innovative and lightweight SR imaging system for mobile phones. M^{3}Cam can ensure high-quality 16× SR image (4× in both height and width) visualization with almost negligible latency. In detail, we utilize an optical image stabilization (OIS) module for lens control and introduce a new modality of data, namely gyroscope readings, to achieve high-precision and compact optical flow estimation modules. Building upon this concept, we design a multi-frame-based SR model utilizing the Swin Transformer. Our proposed system can generate a 16× SR image from four captured low-resolution images in real-time, with low computational load, low inference latency, and minimal reliance on runtime RAM. Through extensive experiments, we demonstrate that our proposed multi-modal optical flow model significantly enhances pixel alignment accuracy between multiple frames and delivers outstanding 16× SR imaging results under various shooting scenarios. *Code and dataset are available at: https://github.com/liangjindeamo-yuer/M3CAM*

CCS Concepts

• **Computing methodologies** → **Image representations**; **Image processing**.

Keywords

Super-resolution system; Optical flow; Mobile camera

ACM Reference Format:
Yu Lu, Dian Ding, Hao Pan, Yongjian Fu, Liyun Zhang, Feitong Tan, Ran Wang, Yi-Chao Chen, Guangtao Xue, and Ju Ren. 2024. M^{3}Cam: Lightweight Super-Resolution via Multi-Modal Optical Flow for Mobile Cameras. In *The 22nd ACM Conference on Embedded Networked Sensor Systems (SENSYS '24), November 4–7, 2024, Hangzhou, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3666025.3699371

1 Introduction

The need for super-resolution (SR) imaging in mobile phone cameras is growing as user demands evolve. Smartphone manufacturers are adding telephoto lenses to meet the desire for clear images of distant subjects. However, the pursuit of ultra-thin designs conflicts with the size of telephoto lenses, limiting SR imaging capabilities. When users zoom into an image post-capture, they often face blurred details and noise, especially with distant scenery. This issue highlights the urgency for SR technology that surpasses the physical limits of smartphone camera hardware.

Deep learning-based SR technology is widely studied today. Single-frame SR (SFSR) models [24, 29, 31, 54] achieve SR from one low-resolution (LR) image. However, SFSR methods require substantial computational resources and may introduce artifacts or excessive smoothing, affecting image quality [8]. Multi-frame SR (MFSR) models [4, 10, 32, 33] create high-resolution images using multiple LR images of the same scene from different positions. These approaches yield better SR results by relying on actual sampled values. However, MFSR models also require significant

*Yu Lu and Dian Ding share the same contribution to this work
[†] Hao Pan is the corresponding author
[‡] Also with Shanghai Key Laboratory of Trusted Data Circulation, Governance and Web3

SENSYS '24, November 4–7, 2024, Hangzhou, China

ACM ISBN 979-8-4007-0697-4/24/11
https://doi.org/10.1145/3666025.3699371

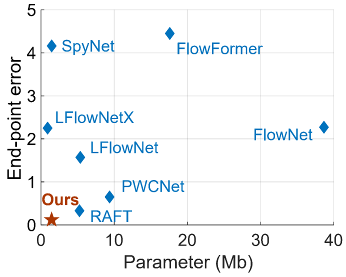


Figure 1: Among existing DNN-based optical flow estimation models, our model achieves the smallest size and best performance simultaneously

	ESRGAN[47]	**DBSR**[4]	**BIPNet**[10]	**EBSR**[33]	**BSRT**[32]	**Ours**
PSNR↑	28.22	35.23	35.26	34.96	35.89	**36.49**
SSIM↑	0.6782	0.8876	0.8603	0.8629	0.8812	**0.8917**
LPIPS↓	0.2704	0.0989	0.0934	0.0945	0.0847	**0.0687**
Frame Num.↓	**1**	14	8	8	12	**4**
onnx. size (MB) ↓	63.8	49.3	25.6	36.7	27.1	**9.33**
RAM (MB)↓	812.34	827.2	753.7	736	721.3	**479.4**
Latency (s)↓	6.65	3.962	9.231	11.58	8.41	**1.39**
Power (J)↓	34.263	22.703	43.337	51.068	39.535	**9.495**

Table 1: Comparison of existing SOTA MFSR systems. All these systems are optimized through ONNX [1] and deployed on various Android smartphones, followed by the measurement of SR imaging quality and on-device inference metrics

computational resources, leading to inference latency on mobile devices.

In this paper, we develop *a lightweight SR system designed for real-time and lightweight SR imaging on mobile devices*. Considering the better performance of MFSR methods, as well as the ease of obtaining multi-frame information of a scene using mobile cameras, we opt for the MFSR technique. Mainstream MFSR methods typically involve the following steps: capturing multiple frames, computing optical flow between reference and adjacent frames, aligning pixels, and ultimately synthesizing an SR image [4, 5, 10, 32, 33]. Among these steps, optical flow estimation is critical, as it directly affects the quality of the resulting super-resolved image. For 16× SR imaging, *i.e.*, 4× resolution along both x and y axes, optical flow calculations with 0.25-pixel precision are needed for accurate alignment. With this precision, at least four frames can be merged to create a single high-resolution image through SR reconstruction.

Existing optical flow models face challenges for smartphone deployment due to computational complexity and accuracy. As shown in Fig. 1, the optical flow network PWCNet [43], used in SOTA MFSR networks, achieves an average endpoint error (a measure of pixel alignment error) of 0.65 pixels. However, its parameters consume up to 72.41% of the total SR system, with a size of 9.37 Mb. Conversely, lightweight networks like SpyNet [40] achieve an average accuracy of 4.2 pixels, insufficient for high-quality SR imaging. Therefore, achieving accurate optical flow with low computational effort, using only RGB information, remains challenging. Researchers have integrated data from non-RGB modalities such as LiDAR [28, 52], infrared cameras [18, 20], and radar [7] to propose multimodal optical flow estimation methods with higher accuracy and robustness. However, these methods require additional modalities (*e.g.*, point cloud) not commonly found in mobile phones, increasing system and model complexity.

In designing M^{3}Cam, we introduce a new modality directly associated with optical flow to enhance problem-solving. This approach aims to achieve two objectives: decreasing model complexity and increasing inference accuracy. Related works [38, 39] have confirmed the strong correlation between lens movement in the Optical Image Stabilization (OIS) camera and optical flow. In contrast, our work leverages lens movement controlled by OIS, combined with visual and gyroscope sensor readings (a new modality), to design a novel, lightweight, and high-performance multimodal optical flow estimation module based on a neural network. To address the challenge of acquiring training data, we synthesized an artificial dataset containing simulated gyroscope readings, multi-frame offset images, and precise optical flow data as ground truth. Using this dataset, we trained a network module with high optical flow estimation accuracy. A comparison between our proposed multimodal optical flow module and the current SOTA optical flow estimation model in terms of model parameter size and optical flow estimation accuracy is shown in Fig. 1. Our model requires only 145 Million parameters, yet achieves a remarkable 0.12 pixel optical flow estimation accuracy. This confirms our approach of introducing a new data modality related to OIS-controlled lens motion, breaking the model size and accuracy trade-off in optical flow estimation.

Building upon this multimodal optical flow module, we propose M^{3}Cam, a real-time 16× SR system that utilizes the Swin Transformer framework to merge multiple aligned LR images. We implement a prototype of our M^{3}Cam system on Android smartphones, that realizes SR imaging by directly processing multiple frames captured in RAW format. Unlike PNG, JPG, or other compressed image formats, RAW files offer a significant advantage as they are uncompressed, meaning they retain all the data directly from the camera's CMOS sensor without any loss. Our system is capable of providing near real-time SR imaging when users wish to examine captured image details of a distant scene, such as when double-tapping the screen to zoom in on a localized area. Our prototype achieves 16× SR enhancement, upscaling 112 × 112 images to 448 × 448, with an average processing time of 1.39 seconds on the tested smartphones' CPUs. The peak memory usage of M^{3}Cam is about 479.4 MB, facilitating near real-time operation on mobile devices. Tab. 1 shows the experimental comparison analysis between M^{3}Cam and the existing mainstream MFSR systems. The results demonstrate that our system significantly surpasses existing SR systems in terms of model lightweight and imaging quality. Specifically, M^{3}Cam achieves a reduction in inference latency of up to 88.0% and a decrease in model size (i.e., the .onnx file) by up to 85.4%. Importantly, our system delivers superior imaging quality, achieving a Peak Signal-to-Noise Ratio (PSNR) of 36.49, a Structural Similarity Index Measure (SSIM) of 0.8917, and a Learned Perceptual Image Patch Similarity (LPIPS) score of 0.0687 for 16× SR imaging. In summary, our proposed M^{3}Cam is lightweight, low-latency, and can be easily integrated into mobile and web applications. Considering the battery capacities of mainstream smartphones are about

5000mAh (equivalent to ~66600 J), our model's inference process is energy-efficient.

The main contributions of this work are as follows:

- We propose a novel multi-modal optical flow estimation module that incorporates lens movement information as new modal data, and achieve a low-computation and high-precision optical flow estimation model.
- Incorporating our proposed multimodal optical flow model, we propose M^{3}Cam, a lightweight SR network based on the Swin Transformer. It can be efficiently deployed on smartphones, achieving real-time inference for 16-fold SR imaging.
- We implement a prototype of our M^{3}Cam and deploy it on various Android smartphones, designed to apply 16-fold SR imaging to specific areas of an image when a user zooms in. We conduct evaluation experiments on upscaling from 112×112 to 448×448, results show that our system can achieve an average latency of just 1.39s and requires 479.4 MB of running RAM, while the power consumption is measured at 9.495 J.

The remainder of this paper is organized as follows: In Sec. 2, we introduce the principles of OIS and lens motion and verify the feasibility of multi-modal optical flow estimation. The design of the proposed lightweight SR system is detailed in Sec. 3, including the multimodal optical flow module and the SR model. Sec. 4 outlines the experimental setup, system evaluation, and user study. In Sec. 5, we discuss the effects of different factors on SR performance during shooting. In Sec. 6, we present the related works about the mobile SR system, visual optical flow, and MFSR methods. Finally, the conclusion of M^{3}Cam is presented in Sec. 7.

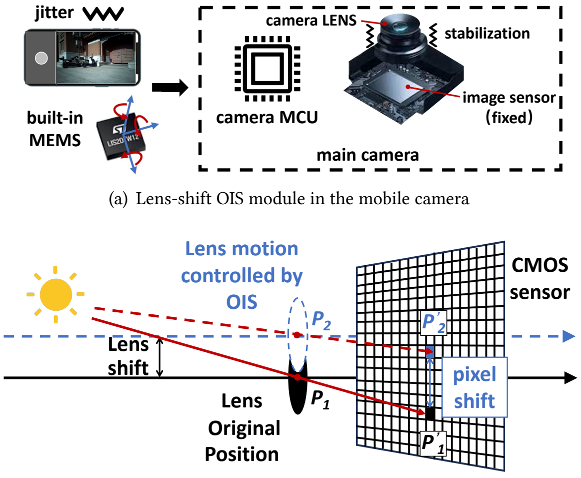
2 PRELIMINARY STUDY

2.1 OIS and Lens Motion

Optical Image Stabilization (OIS) is a mechanical technique that uses an Inertial Measurement Unit (IMU) sensor to monitor and counteract camera shake during image capture by adjusting the lens position with a Voice Coil Motor (VCM) actuator [6]. As shown in Fig. 2(a), the image sensor remains stationary at the base of the camera module. In lens-shift OIS systems, any lens movement alters the optical path to the image sensor, stabilizing the captured image. This prevents distortion from pixel blurring or unwanted artifacts, enabling the production of a nearly perfect digital replica.

2.1.1 Lens Movement via Acoustic Injection. Considering that there are currently no APIs in smartphones that can directly control the lens movement in OIS, we refer to previous relevant studies [38, 39] and use the method of acoustic injection to control the readings of the gyroscope, specifically the 3-axis MEMS gyroscope readings, to further control lens movement. Specifically, by generating an acoustic sinusoidal signal close to the resonance frequency of the gyroscope's sensing mass, which mainly ranges from 18KHz to 30KHz [13] and is inaudible and harmless to human ears, the readings of the MEMS gyroscope can be altered. As a result, the OIS actuator can achieve regular and stable lens movement, such as translational movement along the x/y axis, based on the altered gyroscope readings.

2.1.2 Lens Movement and Optical Flow Correlation. The lens movement can similarly affect the optical flow, almost identical to camera motion. As depicted in Fig. 2(b), a pinhole camera model is utilized to succinctly delineate the relationship between lens shifts and pixel shifts, *i.e.*, optical flow information. When the lens undergoes a displacement δh in one dimension, transitioning from P_1 to P_2, the image of the light source also shifts from pixel P_1' to pixel P_2' with a displacement of δd. Based on similar triangles, we derive the following formula: $\frac{\delta h}{\delta d} = \frac{L}{L+f}$, where L denotes the depth of the light source, and f represents the distance from the lens to the image sensor. More generally, we can leverage regular lens movement to acquire multiple frames within a specified period. If we define the motion offset of the lens on the x/y axis as δh_{x_1},δh_{x_2},δh_{x_3},...,δh_{x_m} and δh_{y_1},δh_{y_2},δh_{y_3},...,δh_{y_m} and displacement of light source in the imaging plane as δd_{x_1},δd_{x_2},δd_{x_3},...,δd_{x_m} and δd_{y_1}, δd_{y_2},δd_{y_3},...,δd_{y_m}, then they should satisfy the following equations:



(a) Lens-shift OIS module in the mobile camera

(b) Pixel shift caused by OIS-controlled lens motion

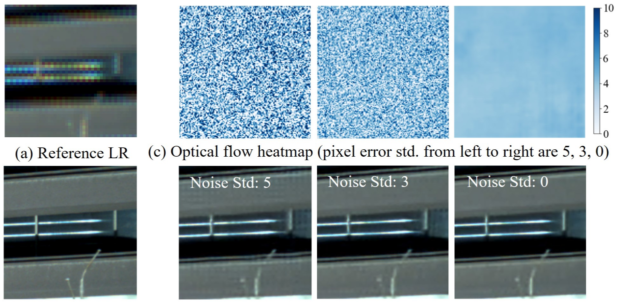
Figure 2: (a) OIS module controls the lens based on the built-in gyroscope readings to prevent blurring; (b) Difference in pixel coordinates of the same light source projected onto two frames, *i.e.*, optical flow, is linearly related to the distance of the lens movement

$$\begin{aligned}\frac{\delta h_{x_1}}{\delta d_{x_1}} = \frac{\delta h_{x_2}}{\delta d_{x_2}} = \frac{\delta h_{x_3}}{\delta d_{x_3}} = \cdots = \frac{\delta h_{x_m}}{\delta d_{x_m}} = \frac{L}{L+f} \\ \frac{\delta h_{y_1}}{\delta d_{y_1}} = \frac{\delta h_{y_2}}{\delta d_{y_2}} = \frac{\delta h_{y_3}}{\delta d_{y_3}} = \cdots = \frac{\delta h_{y_m}}{\delta d_{y_m}} = \frac{L}{L+f}\end{aligned} \tag{1}$$

In addition, there is a strict positive correlation $\mathcal{F}$ between lens motion and MEMS gyroscope readings:

$$\delta h_x = \mathcal{F}(\theta_x(T)), \delta h_y = \mathcal{F}(\theta_y(T)) \tag{2}$$

From Eq.1 and Eq.2, we conclude that for cameras supporting OIS, the gyroscope readings after acoustic injection are modal data directly related to the optical flow. Actually, OISSR [39] has utilized gyroscope readings to enhance optical flow estimation. However, it only achieved 4× SR and lacked validation for real-time deployment on mobile devices. Therefore, this prompts us to incorporate gyroscope readings as a new modality to design a multimodal optical flow estimation module based on a neural network, aiming to achieve both light weight and high accuracy.



(a) Reference LR (c) Optical flow heatmap (pixel error std. from left to right are 5, 3, 0)

(b) SR ground truth (d) Generated SR from multiple frames with optical flow results in (c)

Figure 3: Impact of optical flow errors (standard deviation) on the quality of 16× SR. With sufficiently accurate optical flow estimation, synthesizing SR images from multi-frame, varied camera positions can yield near-identical results to the HR ground truth

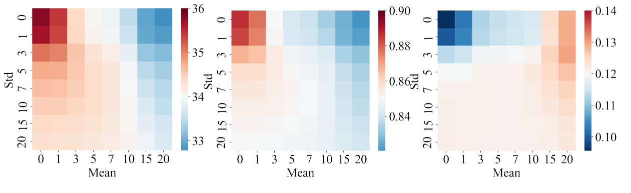
2.2 Optical flow estimation for MFSR

2.2.1 Effect of optical flow on SR imaging. The optical flow estimation module calculates the optical flow vectors between a reference LR frame and multiple offset LR frames and aligns them, which is an important process for generating SR outputs. However, even a pre-trained optical flow network struggles to predict inter-frame optical flow information with sub-pixel level accuracy.

Inaccurate optical flow estimation can cause blurred or misaligned images when frames are merged, significantly lowering the quality of the SR image. We examine the impact of optical flow estimation error on high-resolution image generation using the synthetic dataset described in Sec. 4.1.1. We tested different SR models, including DBSR [4], BIPNet [10], EBSR [33], and BSRT [32], by skipping their optical flow estimation networks and inputting erroneous optical flow vectors. The results are shown in Fig.3 and imaging quality metrics (PSNR, SSIM, LPIPS) are shown in Fig.4. We find that higher optical flow estimation accuracy leads to better synthesized SR images.

2.2.2 Analysis of optical flow model size. Designing an optical flow estimation module with sub-pixel accuracy is challenging. When estimating optical flow for multiple frames, it must accurately estimate pixel motion under different lighting conditions and scenarios, handle numerous LR inputs, and ensure consistency and accuracy of the estimates. Moreover, it must withstand factors like noise and lighting changes that can affect image quality. Balancing these needs complicates the module's design. We analyze the size of SOTA MFSR models and their optical flow modules (Tab. 2). It is observed that the high-precision optical flow estimation method, PWCNet [43], used in DBSR [4] and MFIR [5], leads to substantial model parameters, accounting for approximately 60-70% of the total SR model size. Additionally, simplifying the optical flow estimation module with SpyNet [41] in EBSR [33] and BSRT [32] significantly reduces SR model size. However, this reduction results in decreased optical flow accuracy, requiring additional network modules for compensation.

We identify a contradiction in the optical flow estimation module between model complexity and accuracy, primarily due to pixel displacement caused by uncertain subtle shaking in multi-frame images from handheld photography. Optical flow modules relying solely on visual data require complex network designs to maintain robustness and precision under varying camera motion scenarios. Given the computational constraints of mobile devices, lightweight optical flow models fail to achieve high-performance SR imaging.



(a) PSNR (↑) (b) SSIM (↑) (c) Lpips (↓)

Figure 4: Impacts of optical flow estimation errors (both mean and std.) on the quality metrics of generated SR images

SR Model	**DBSR**[4]	**MFIR**[5]	**EBSR**[33]	**BSRT**[32]
# of Para. (M)	12.94	15.83	9.52	7.06
OF Model	PWCNet	PWCNet	SpyNet	SpyNet
# of Para. (M)	9.37	9.37	1.44	1.44
Pixel error	0.65	0.65	4.16	4.16
OF/SR size ratio (%)	72.41	59.19	15.13	20.39

Table 2: Parameter analysis of optical flow (OF) estimation modules in different SR systems

3 M^{3}Cam DESIGN

3.1 System Overview

Our objective is to develop a lightweight and efficient SR system for mobile cameras. Acknowledging the critical role of accurate optical flow estimation in MFSR systems, our approach diverges from conventional methods that rely solely on visual data. We leverage acoustic signals to guide the OIS lens movement and incorporate gyroscope readings to enhance optical flow estimation.

The workflow of our M^{3}Cam is illustrated in Fig.5. Initially, we capture multiple RAW images along with synchronized gyroscope readings (see Sec.3.2). The proposed multimodal optical flow module (Sec.3.3) then estimates the optical flow between the reference frame and the offset frames. To enhance SR synthesis efficiency and reduce computational load, we introduce a frame filtering module (Sec.3.4) that selects the optimal four frames for final synthesis. The aligned LR images are subsequently fed into the SR model to generate 16× ultra-high resolution images (Sec.3.5). Additionally, we convert the RAW images to RGB format for visualization (Sec.3.6).

3.2 Capturing Multiple RAW Frames

While capturing multiple frames, we use a speaker built in the smartphone to play a high-frequency audio signal, such as a 19.61 kHz sine wave[1], to influence changes in the smartphone's built-in gyroscope readings, which in turn causes the OIS to move the lens. So, multiple LR RAW frames $\Psi = \{I_i | i = 0, 1, 2, \ldots, m\}$ are captured at different lens positions, with concurrent gyroscope readings obtained through sampling. A multimodal optical flow network

[1] The high-frequency acoustic wave is determined by the smartphone's built-in gyroscope mass resonant frequency during initialization. This frequency is identified by playing a sweep-frequency audio signal and analyzing the gyroscope readings.

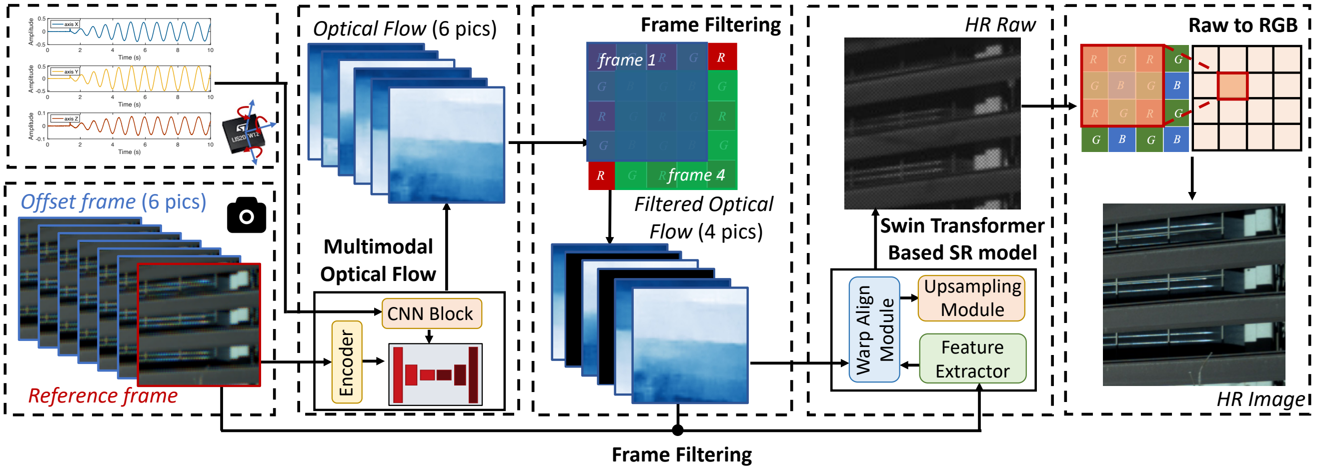


Figure 5: Overview of our designed M^{3}Cam, a lightweight mobile 16×SR system based on the novel multi-modal optical flow estimation module

calculates optical flow $\psi = \{f_i | i = 0, 1, 2, \ldots, m\}$ for each frame relative to a typical reference frame, where $f_0 = \vec{0}$. This new modality of data aids the optical flow module in achieving faster and more precise flow estimations.

It is important to note that the lens movement speed can be modulated by altering the frequency of the audio signal. In well-lit conditions, we often choose quicker lens motion using, for example, a sine wave of approximately 19.60 kHz or 19.62 kHz, completing a full cycle within half a second.

3.3 Multimodal Optical Flow Module

For each pair of the reference frame $I_0 \in \mathcal{R}^{3\times H\times W}$ and a target offset frame $I_i \in \mathcal{R}^{3\times H\times W}$, a convolutional network-based encoder extracts features, transforming the input image into LR dense features $E(I) \in \mathcal{R}^{D\times \frac{H}{8}\times\frac{W}{8}}$. This encoder includes 6 residual blocks: 2 at 1/2 resolution, 2 at 1/4, and 2 at 1/8. Visual similarity is assessed by generating a comprehensive correlation volume for all pairs,

$$C(E(I_0), E(I_i)) \in \mathcal{R}^{\frac{H}{8}\times\frac{W}{8}\times\frac{H}{8}\times\frac{W}{8}}$$
$$C_{h_0,w_0,h_i,w_i} = \sum_{d}^{D} E(I_0)_{d,h_0,w_0} \cdot E(I_i)_{d,h_i,w_i}, \tag{3}$$

The initialized optical flow f_{init} is obtained from the gyroscope data by multiplying a weight matrix $\mathcal{W}$. The correction module is designed to refine the initialized optical flow f_{init} by integrating visual information $C(E(I_0), E(I_i))$ and $E(I_0)$, thereby achieving an initially precise optical flow f_{vis}. Meanwhile, convolutional blocks are employed to extract intricate features f_{gy} from the gyroscope data. Ultimately, f_{gy} and f_{vis} are merged within the Unet framework to yield the final optical flow, denoted as f_i.

The optical flow calculated by our multimodal optical flow network, as depicted in Fig. 6, is defined as $\{f_i | i = 0, 1, 2, \ldots, m\}$ for each image frame. From Eq. 1, the optical flow $(\delta d^A_{x_i}, \delta d^A_{y_i}, \delta d^A_{x_j}, \delta d^A_{y_j})$ of a single light source A in two frames satisfies the constraint relationship: $\frac{\delta d^A_{x_i}}{\delta d^A_{x_j}} = \frac{\delta h_{x_i}}{\delta h_{x_j}}$ and $\frac{\delta d^A_{y_i}}{\delta d^A_{y_j}} = \frac{\delta h_{y_i}}{\delta h_{y_j}}$. Then for light source B, we still have: $\frac{\delta d^B_{x_i}}{\delta d^B_{x_j}} = \frac{\delta h_{x_i}}{\delta h_{x_j}}$ and $\frac{\delta d^B_{y_i}}{\delta d^B_{y_j}} = \frac{\delta h_{y_i}}{\delta h_{y_j}}$. Therefore, the following equation holds:

$$\frac{\delta d^A_{x_i}}{\delta d^A_{x_j}} = \frac{\delta d^B_{x_i}}{\delta d^B_{x_j}} \quad \frac{\delta d^A_{y_i}}{\delta d^A_{y_j}} = \frac{\delta d^B_{y_i}}{\delta d^B_{y_j}} \tag{4}$$

Utilizing the aforementioned equation, we can employ the following objective function to optimize and train our optical flow network:

$$\min \sum_{i=2}^{m} (std(\frac{f_{x_i}}{f_{x_1}}) + std(\frac{f_{y_i}}{f_{y_1}})) \tag{5}$$

3.4 Frame Filtering Module

Given the captured frames $\Psi = \{I_i | i = 0, 1, 2, \ldots, m\}$ during lens movement controlled by acoustic injection and their corresponding optical flow $\psi = \{f_i | i = 0, 1, 2, \ldots, m\}$, combining them all in a neural network for feature extraction and fusion incurs significant computational overhead and resource wastage. Furthermore, as outlined in Sec. 2.1, specific images captured through random sampling might exhibit approximate lens positions and optical sampling of the scene due to the sinusoidal acoustic signal-induced periodic lens shifts. For example, we measure the information of one image in terms of 2D entropy [51]. Considering two images I_1 and I_2 with dimensions H in height and W in width, we define $F(i, j)$ as the occurrence frequency of a pixel value i along with the mean value j of its surrounding region. Additionally, we define the probability that $F(i, j)$ occurs in images I_1,I_2 as p_{ij} and q_{ij}. We employ KL divergence [45] to quantify the variability between two images:

$$P_{I_1} = \{p_{ij}\} = \{\frac{F_{I_1}(i,j)}{H\times W}\},\ P_{I_2} = \{q_{ij}\} = \{\frac{F_{I_2}(i,j)}{H\times W}\}$$
$$D_{KL}(I_1||I_2) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{6}$$

When the image sensor has an approximate sampling of the current scene in both images I_1,I_2, $D_{KL}(I_1||I_2) \approx 0$. Consequently, for the series of images captured by the acoustic injection capturing module, we apply a frame filtering module to filter the set of

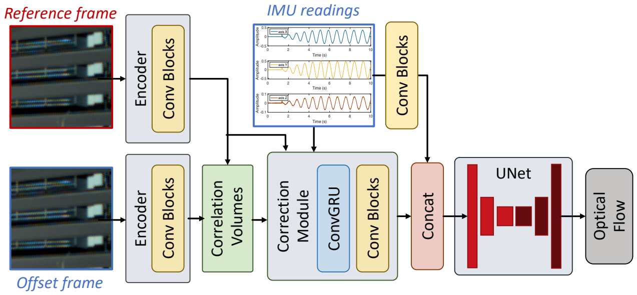


Figure 6: Multimodal optical flow estimation model

valid images $\Phi = \{I_i | i = 0, 1, \ldots, n-1\}$, such that KL divergence $D_{KL}(I_i || I_j) > \mathcal{B}$ for any $I_i \in \Phi$ and $I_j \in \Phi$ ($\mathcal{B}$ is the threshold determined by the dataset). Meanwhile, supplying the filtered frames to the SR network alleviates the computational load on the network.

Subsequently, we elaborate on the methodology for carrying out efficient frame filtering. Considering the regular distribution of camera image sensors, such as Bayer arrays [35], it is essential to take into account the periodicity of sensor sensing. To simplify, if the optical flow of a frame relative to the reference frame is exactly $2k$ pixels ($k \in \mathcal{N}$) in both x-/y-axis, then the information captured in the two frames is the same. Therefore, we use the mode operation to normalize the optical flow. Furthermore, it is crucial to consider that the light source at the image's edge might not be visible in the reference frame, rendering the optical flow invalid for that particular pixel point. For each burst frame, we obtain the average optical flow

$$\overline{\psi} = \{\overline{f_i} = (Mean(f'_{ix})mod2, Mean(f'_{iy})mod2)\} \tag{7}$$

with some of the edges cropped out. With the neural network taking n frames as input, we derive the set Φ consisting of valid frames with a potential of n elements by clustering the average optical flow set $\overline{\psi}$ into n clusters using the K-means clustering algorithm [34]. Proceeding further, we select the data points that are closest to the cluster centroid within each cluster to serve as representatives for the valid frames set Φ. It is crucial to underscore that when a cluster includes the reference frame (i.e.,$\overline{f_{ix}} = \overline{f_{iy}} = 0$), we promptly assign the reference frame as the representative.

3.5 Super Resolution Network

In this section, we describe our 16× SR system. Given the valid frame set $\Phi = \{I_i | i = 0, 1, 2, \ldots, n-1\}$ and their corresponding set of optical flows $\phi = \{f_i | i = 0, 1, 2, \ldots, n-1\}$, the objective of our model is to leverage the shifted complementary information from different images to reconstruct a SR image. Each image $I_i \in \mathbb{R}^{4 \times \frac{H}{2} \times \frac{W}{2}}$ is 4-dimensional data arranged in channels of RGGB. The overview of our architecture is shown in Fig. 7.

3.5.1 *Encoder.*

The encoder module transforms each frame I_i into a feature representation e_i with significant depth. The RAW data captured by the camera is packed along each 2×2 block of RGGB in the Bayer array, resulting in the formation of input frames I_i across the four channels. To accomplish high-dimensional feature extraction and encoding, we first use a convolutional block to expand the dimensionality of the features from 4 dimensions to n_{fea} (default is 96) dimensions. Considering that Swin Transformer employs a window mechanism that divides the image into non-overlapping windows, performing self-attention within each window independently, we employ the Swin Transformer block to systematically extract features essential for subsequent image alignment and fusion. Finally, we use a Conv-Pixelshuffle module to convert the extracted features from $n_{fea} \times \frac{H}{2} \times \frac{W}{2}$ to $n_{fea} \times H \times W$. The resulting n_{fea}-dimensional encoding $e_i \in \mathbb{R}^{n_{fea} \times H \times W}$ effectively achieves a comprehensive embedding of the input image.

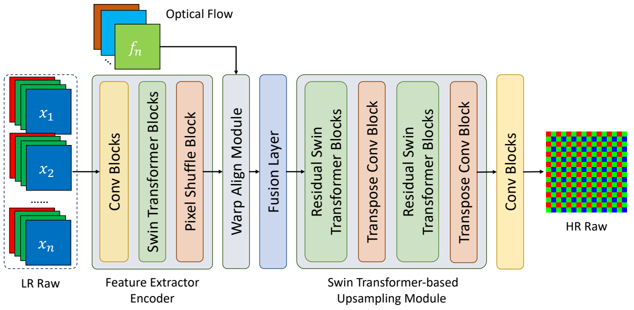


Figure 7: Detailed framework of Swim Transformer-based 16 × SR network

3.5.2 *Alignment Module.*

To attain a proficient fusion of multiple frames, it is imperative to align the information beforehand from the encoder. Considering that we have obtained the optical flow $f_i \in \mathbb{R}^{2 \times H \times W}$ from the multimodal optical flow module in advance, we utilize this priori bias information to align the multi-frame information to a selected reference frame. For simplicity, we set e_0 as the cropped reference frame and use the warp function to align e_i to e_0: $\widehat{e_i} = warp(e_i, f_i)$. Specially, $\widehat{e_i} = e_i$ for $i = 0$. Through the meticulous alignment offered by the multimodal-based optical flow, we constrain the optical sampling of frame images sharing the same light source to pixel positions with minimal deviations, thereby enabling the fusion module to streamline feature fusion within compact pixel blocks.

3.5.3 *Fusion Module.*

The fusion module integrates information from the individual image embedding $\widehat{e_i}$ to create a unified feature embedding, denoted as $\widetilde{e}$. We use convolutional blocks to perform the fusion of multi-frame image embedding $\widehat{e_i}$. The multi-frame image embeddings $\widehat{e_i}$ are concatenated along the channel dimension, resulting in $\widehat{e} \in \mathbb{R}^{n \cdot n_{fea} \times H \times W}$. Next, we utilize a convolutional neural network to compress the joint image features $\widehat{e}$ from $n \cdot n_{fea}$ dimensions to n_{fea} dimensions to obtain the fused feature map $\widetilde{e} \in \mathbb{R}^{n_{fea} \times H \times W}$.

3.5.4 *Upsampling Module.*

The upsampling module generates the final high-resolution RAW image output from the fused feature $\widetilde{e}$. We also employ the Swin Transformer as the upsampling module. Swin Transformers are known to be versatile backbones in various computer vision applications, demonstrating excellent performance in image classification and instance segmentation. It can efficiently handle tasks by increasing the dimensionality of positional features and reducing the image feature's height and width. Within this backbone, each Swin Transformer block maintains the same input and output shape, with feature size changes achieved through a Patch Merging layer located between the Swin Transformer blocks. Patch Merging layer converts the shape of the input matrix from $C{\times}H{\times}W$ to $2C{\times}\frac{H}{2}{\times}\frac{W}{2}$. Thus, we modify this layer by replacing the Patch Merging layer with a *transposed convolutional layer*, which upsampling the input tensor (i.e. changing the $C \times H \times W$ tensor to $C{\times}2H{\times}2W$). Meanwhile, to address the problem of degradation that tends to occur during training, we use the *residual Swin Transformer block* in place of the original Swin Transformer block. After two rounds of upsampling by the transposed convolutional layer, we can get a feature map of size $4H \times 4W$. The upsampled feature map is then passed through another set of convolutional blocks to obtain the high-resolution Raw image $y \in \mathbb{R}^{1\times 4H\times 4W}$. Our model's RAW2RAW input/output design eliminates the need for additional structures to learn RAW to RGB image processing, significantly reducing the neural network's size when using PyTorch.

3.6 RAW to RGB Conversion

Mobile cameras typically use an integrated RAW2RGB module to convert RAW images to RGB format. This module involves processes like Bayer pattern demosaicing, white balance, color space mapping, noise reduction, tone mapping, and color manipulation[36]. Accessing this pipeline is difficult due to proprietary techniques in camera hardware, unique to each manufacturer[23]. For quality assessment (PSNR[49], SSIM[48], LPIPS[53]) in our experiments, we use a simulated software pipeline based on[23]. This pipeline processes the RAW images, *i.e.*, .dng format files, representing the camera's unprocessed CMOS sensor response, produced by our neural network. We generate RGB images using the burst Bayer array and capturing parameters from the *.dng* file for visual evaluation, and the details are shown in Alg. 1.

Algorithm 1 Raw to RGB Processing Algorithm

```
Input: Bayer image with color space of 10 bits $I$, metadata of the
    original image including maximum and minimum pixel values,
    black level $b_l$, white balance $w_b$, bright factor $b_f$ and gamma
    correction $\gamma$
Output: A RGB image $I_{rgb}$ with color space of 8 bits
  Convert from 10 bits to 8 bits by directly dividing 4
  Normalize the image $I$ and black level $b_l$ to [0., 255.]
  Subtract the black level $b_l$ from $I$
  Demosaicing: Convert $I$ from Bayer RGGB to RGB
  Apply white balance $w_b$ and bright factor $b_f$
  Perform gamma correction and smooth the image
  Adjust colors of the image independently
  return RGB image $I_{rgb}$
```

4 EVALUATION

4.1 Experimental Setup

We develop a M^{3}Cam prototype for Android smartphones with a camera supporting a lens-shift OIS module and assess the performance of our proposed multimodal optical flow estimation module, 16× SR imaging quality, and system inference in mobile deployment scenarios.

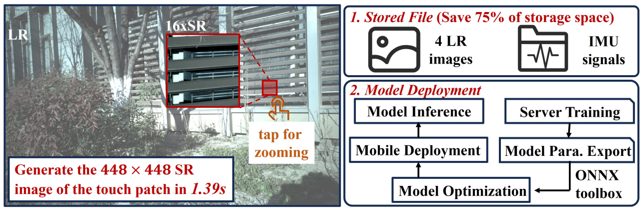


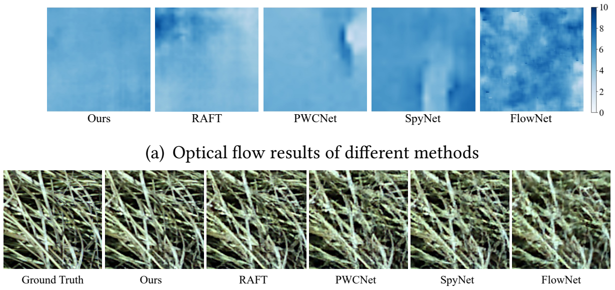
Figure 8: M^{3}Cam performs model training at the server and is deployed on mobile devices to achieve real-time SR imaging

4.1.1 Dataset. **Real-world Dataset Collection:** When collecting training data, we hold the smartphone by hand. It is important to note that if significant handshaking occurs while capturing LR images, we will discard the data collected at that time. Considering that capturing multiple frames of images only requires 0.5 to 1 second, maintaining hand stability is relatively easy. We use a Xiaomi 11 Pro to capture 500 training sets of high-resolution RAW images (*.dng* format) with a resolution of 4096×3072 of different scenes, 20 images per set. The details of taking multiple images are described in Sec. 3.2. The camera settings in good light are set as an aperture of f/1.9, a shutter speed of 1/1012 seconds, and a brightness value of 12.8. In bad light conditions, we increase the shutter speed to 1/500 seconds and others remain unchanged. We take the first frame as the reference frame and crop 48 subfigures sequentially of $1 \times 448 \times 448$ resolution to get the SR ground truth. Next, we downsample the cropped subfigures and pack them along the RGGB channel to get a $4 \times 56 \times 56$ LR image,*e.g.*, $\Psi = \{I_i | i = 0, 1, 2, \ldots, 19\}$ which includes one reference frame and 19 offset frames. Meanwhile, to make the dataset directly applicable to the training of the SR neural network in Sec. 3.5, we input these LR RAW images and gyroscope readings into the multimodal optical flow module to obtain the optical flow $\psi = \{f_i | i = 0, 1, 2, \ldots, 19\}$. Ultimately, we partition the collected 24, 000 subfigures into training, cross-validation, and test sets using a random split ratio of 8 : 1 : 1.

Synthetic Dataset Generation: We produce a synthetic dataset with ground truth optical flow information. We select a cosine function and randomly sample it to obtain a series of image offsets $\mathcal{D} = \{(d_{x_i}, d_{y_i}) | i = 0, 1, 2, 3, \ldots, m\}$ as the synthesized gyroscope data. Next, we apply this offset information in the form of affine transformations to the reference image of the above real-world dataset to obtain multi-frame images. Eventually, we crop and downsample each image in the same way as in a real-world dataset. By performing the above operations, we obtain the synthetic dataset, which has the optical flow information $\mathcal{D} = \{(\frac{d_{x_i}}{4}, \frac{d_{y_i}}{4}) | i = 0, 1, 2, 3, \ldots, m\}$ as ground truth. This dataset is used in Sec. 2.2 to evaluate the effect of optical flow errors on imaging results.

4.1.2 SR Model Training. Our proposed M^{3}Cam is trained on one NVIDIA TESLA V100 for 1000 epochs with a batch size of 4. The optimizer is Adam with a learning rate of 0.0001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, a weight decay of 0. A learning rate scheduler is used to decay the learning rate of each parameter group by $\gamma = 0.25$ on $300th$ epoch and $800th$ epoch.

4.1.3 Mobile Deployment and On-device Test. We present the pipeline of our mobile deployment in Fig. 8. In detail, we export our proposed SR model weights trained on the server and utilize the Open



(a) Optical flow results of different methods

(b) SR imaging results based on different optical flow methods

Figure 9: Comparison of our proposed multi-modal optical flow estimation model versus others and SR imaging results based on optical flow outcomes

	RAFT[44]	PWCNet[43]	SpyNet[41]	FlowNet[9]	Ours
EPE↓	0.33	0.65	4.16	2.27	**0.12**
LAT (s)↓	0.76	0.84	0.42	1.35	**0.19**

Table 3: Comparison of our proposed optical flow module with other SOTA methods. EPE stands for endpoint (pixel) error, LAT stands for latency

	ResNet [19]			Swin Transformer [30]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RAFT[44]	31.03	0.8324	0.147	35.87	0.8841	0.0692
PWCNet[43]	31.09	0.8307	0.151	35.37	0.8794	0.0763
SpyNet[41]	30.84	0.8215	0.159	35.14	0.8687	0.0839
FlowNet[22]	30.15	0.8115	0.165	33.61	0.8335	0.1219
Ours	**31.51**	**0.8401**	**0.132**	**36.49**	**0.8917**	**0.0687**

Table 4: Ablation study of the selected Swin Transformer and ResNet as merging networks with various optical flow estimation methods

Neural Network Exchange (ONNX) [1] toolbox to optimize the entire model weights to be compatible with the hardware of mobile devices. The optimized model is then deployed on Android smartphones. We deploy the proposed M^{3}Cam on five different Android smartphones, and the camera parameters are kept at default parameters. We test the SR imaging quality and on-device resource characteristics, and the specific phone models can be found in Tab. 7. Since our SR system supports real-time inference, users can save only 4 frames of LR images along with the corresponding gyroscope readings, saving up to 75% of storage space.

4.2 Performance of Optical Flow Estimation and SR Imaging

4.2.1 Multi-modal Optical Flow Module Performance. We initially assess the efficacy of our proposed optical flow estimation module, which integrates gyroscope and image fusion. Each image alignment technique is used on a sequence in the dataset to determine pixel offset information. We compared SOTA optical flow modules, including RAFT [44], PWCNet [43], SpyNet [41], and FlowNet [22]. Multi-frame image fusion was then performed using our proposed SR network in M^{3}Cam. As Fig. 9(a) illustrates, the gyroscope and image fusion-based optical flow module accurately extracts pixel offsets between LR images. Conversely, vision-only optical flow modules struggle to avoid substantial local errors. In Fig.9(b), we showcase how different optical flow methods affect the quality of 16× SR images. Optical flow derived from gyroscope and image fusion notably enhances SR image quality, particularly in the clarity of high spatial frequency details. Tab. 3 confirms that incorporating gyroscope data into optical flow estimation for LR images greatly improves the visual quality of the resulting SR images. Additionally, our model demonstrates remarkably low latency, at just 0.19s, significantly outperforming other SOTA methods in efficiency.

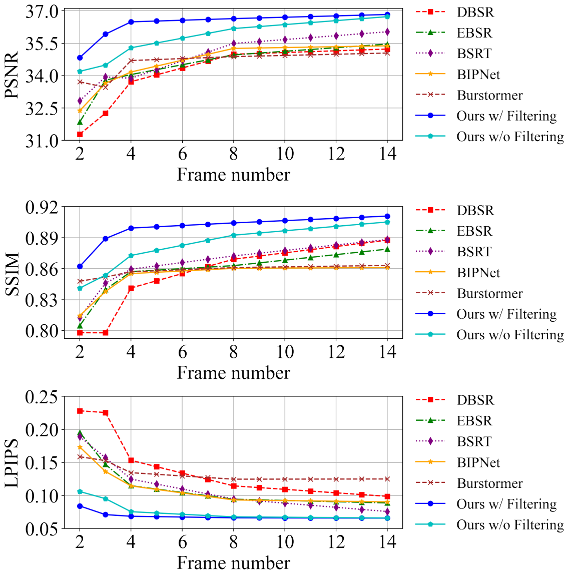
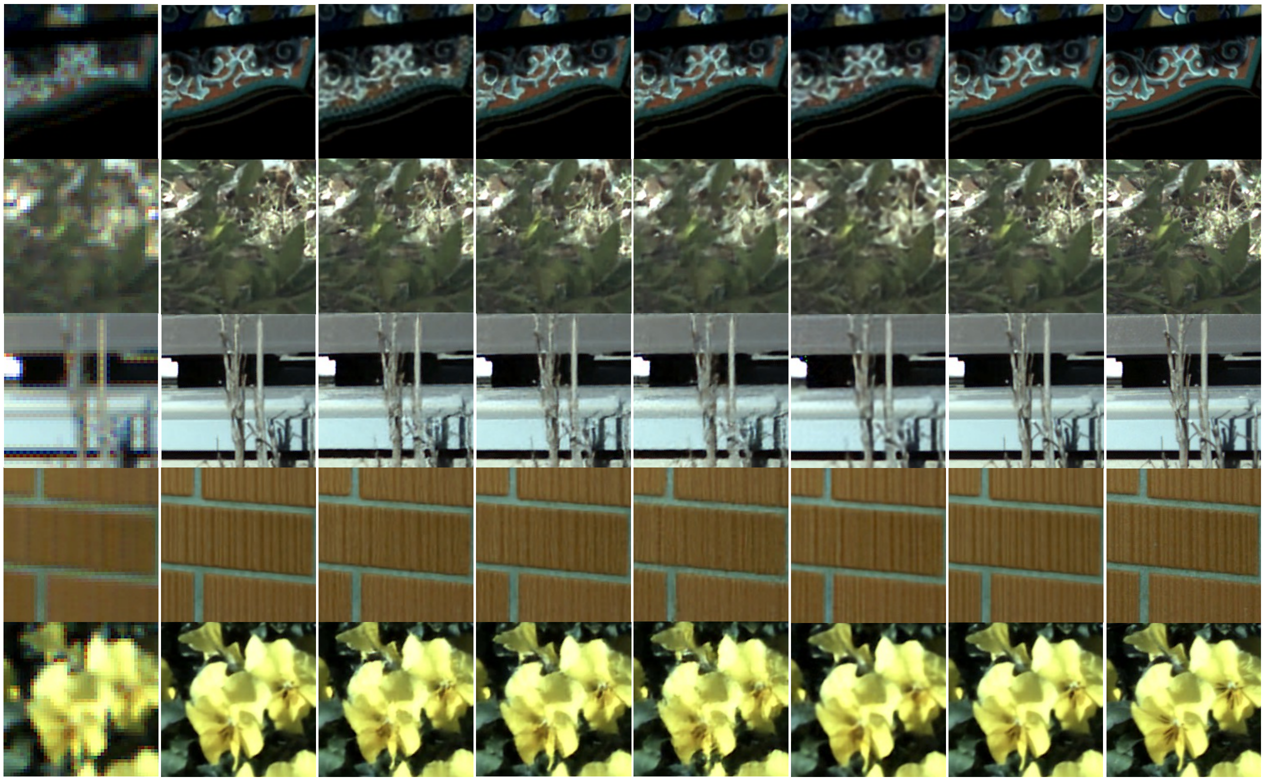


Figure 10: Impact of the number of LR frames on the quality of synthesized 16× SR images

4.2.2 SR imaging performance. **Comparison of Merging Network.** We conduct an ablation study on the module for synthesizing SR images from aligned multi-frame LR images. Specifically, we compare the popular neural network (ResNet50) with the Swin Transformer we are using as the fusion network. We combine these with different optical flow estimation modules and compare the quality metrics of the SR imaging, as shown in Tab. 4. The results indicate that the Swin Transformer performs significantly better in the task of fusing multiple LR images to generate an SR image.

Number of Merged LR Frames. We employ a frame filtering module that selects specific frames from consecutively captured images for merging, reducing information redundancy. Thus, we examine the impact of different quantities of LR images on the final super-resolution image quality, with frame numbers ranging from 2 to 14. As shown in Fig. 10, we find that by using only 4 LR frames for merging, we can almost achieve the optimal visual quality of 16× SR images, comparable to the quality obtained from merging up to 14 frames. This is thanks to our designed, KL scatter-based frame filtering module, which helps us to reduce the processing of redundant information.

RAW-format SR Imaging Performance. In this experiment, we evaluate the performance of different SR networks, including BSRT [32], DBSR [4], EBSR [33], BIPNet [10], and Burstormer [12]. These methods all have the same input and output format, *i.e.*, RAW format (*.dng*). The SR imaging visualization results are shown in



(a) LR image (b) BSRT (c) DBSR (d) EBSR (e) BIPNet (f) BurstFormer (g) Ours (h) HR image

Figure 11: End-to-end imaging visualization comparison of our proposed M^{3}Cam with other SOTA 16× MFSR systems

Metrics	PSNR↑	SSIM↑	LPIPS↓	Para. # (10^6)↓	Latency (s) ↓	RAM (MB) ↓	onnx. (MB)↓	Frame #↓	Power (J)↓
BSRT[32]	35.89	0.8812	0.0847	7.06	8.41	721.3	27.1	12	39.535
DBSR[4]	35.23	0.8876	0.0989	12.94	3.96	827.2	49.3	14	22.703
EBSR[33]	34.96	0.8629	0.0945	9.52	11.58	736	36.7	8	51.068
BIPNet[10]	35.26	0.8603	0.0934	6.67	9.23	753.7	25.6	8	43.337
Burstormer[11]	34.88	0.8610	0.1248	2.49	N/A	N/A	N/A	8	N/A
Ours	**36.49**	**0.8917**	**0.0687**	**2.17**	**1.39**	**479.4**	**9.33**	**4**	**9.495**

Table 5: Comparative analysis of 16× SR imaging quality and on-device inference for various RAW-format MFSR systems. These parameters are measured with Xiaomi 11 Pro by enhancing cropped region from 112×112 pixels to 448×448 pixels. Our system can synthesize 16× SR images using only 4 frames and achieve, or even surpass, the results of other SOTA SR methods. Notably, our system is more lightweight, less computationally intensive, and energy-efficient [2]

Fig. 11, while the experimental results in both the SR imaging quality metrics and on-device inference metrics are shown in Tab. 5. Based on the high-precision optical flow derived from gyroscope readings and image fusion, the output SR images produced by M^{3}Cam exhibit visual quality that is comparable to or exceeds that of SOTA MFSR systems. The performance of our M^{3}Cam mobile deployment on the Xiaomi 11 Pro is shown in Tab. 5. Thus, our proposed M^{3}Cam is highly lightweight and supports real-time SR imaging with a tap zoom area (*i.e.*, a 112×112 pixel region). The performance of mobile deployments of other SOTA MFSR systems and various smartphones will be evaluated in the next section.

4.2.3 Across different smartphones. We evaluate the on-device SR inference performance of various systems across different testing Android smartphones with OIS modules, including Xiaomi 11 Pro, Redmi K40S, Xiaomi Mix 4, Xiaomi 10 and Redmi Note 12 Pro. We first test the performance of the experimental mobile phones using ANTUTU [3], which includes mainly four indicators: CPU, GPU, MEM, and UX. The results are shown in Tab. 6. Specifically, CPU, GPU, and MEM denote the mobile phone’s CPU performance, 3D performance, and RAM performance, respectively, while the UX indicator integrates data security, data processing, image processing, and I/O performance. SUM represents the total score of the test phone across the four aforementioned areas. Among the smartphones selected for our experiments, the Xiaomi 11 Pro demonstrates the strongest neural network inference capabilities.

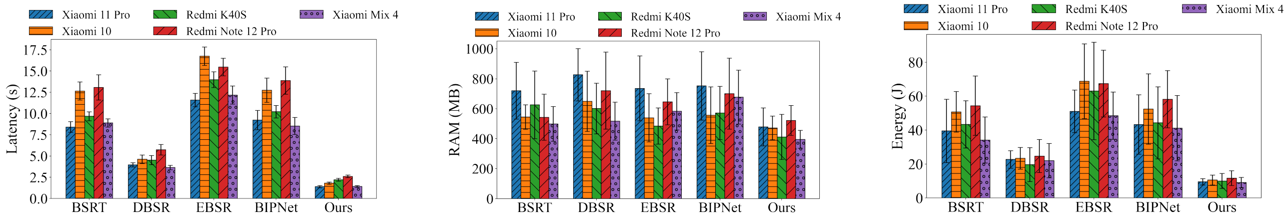
[2] Burstormer relies on the Enhanced Deformable (EDA) Alignment module to elastically deform local features in LR images for improved spatial alignment. However, due to the lack of mobile library support in the EDA module, Burstormer cannot be easily deployed on mobile devices.

Smartphones	CPU↑	GPU↑	MEM↑	UX↑	SUM↑
Xiaomi 11 Pro	177112	198164	138905	167166	681347
Redmi K40S	186205	172761	111705	152525	623196
Xiaomi Mix 4	171285	243559	114380	117454	646678
Xiaomi 10	164215	202188	108087	91243	565733
Redmi Note 12 Pro	157656	144272	89794	134706	526428

Table 6: Comparison of computation micro benchmark tests on the test smartphones

Smartphone	w/o high frequency noise			with high frequency noise		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Xiaomi 11 Pro	36.49	0.8917	0.0687	36.29	0.8823	0.0732
Redmi K40S	36.91	0.8732	0.0733	35.69	0.8734	0.0698
Xiaomi Mix 4	35.89	0.8823	0.0727	36.19	0.8763	0.0726
Xiaomi 10	36.34	0.8756	0.0671	35.93	0.8814	0.0758
Redmi Note 12 Pro	36.74	0.8723	0.0693	35.83	0.8752	0.0703

Table 7: Impact of the high-frequency noise near the testing phones on the SR performance



(a) Latency

(b) RAM

(c) Energy

Figure 12: On-device inference metrics for our M^{3}Cam when deployed across various mobile devices

We next test the mobile deployment performance across different smartphones and SR models. For different SR models, the phone performs SR inference 100 times on different images of the same size 112 × 112 with no other applications running. We first report the inference latency, which is obtained by logging CPU occupancy time with *CPU profiler* toolbox [15] in the *Android Studio*. Compared to the mainstream MFSR methods, M^{3}Cam reduces the latency by 64.91% to 88% and storage by 63.55% to 81.08%. Next, we use the *Memory profiler* [16] to record the RAM information during the on-device inference. The results show that compared to other MFSR methods, M^{3}Cam reduces running memory overhead by 33.53% to 42.04%. In terms of power consumption during the SR inference, we utilize the Battery Historian [14], a tool to inspect battery-related information and events on an Android device. Then the energy overhead of our M^{3}Cam and other SR systems can be calculated based on the battery information before and after the SR inference. The proposed lightweight system consumes only 9.495 J of energy in a single run, which is 58.2% to 81.4% lower than the mainstream MFSR approach and 72.3% to 89.6% lower than the mainstream SFSR approach. As the results have shown, after mobile deployment, M^{3}Cam significantly outperforms other SR systems regarding latency, energy consumption, and memory usage. Meanwhile, the detailed imaging quality of the on-device M^{3}Cam inference measured on different mobile devices are shown in the right part of the Tab. 12. Thus, we can conclude that when deploying M^{3}Cam on the different smartphones, we can achieve close 16× SR performance in terms of both the SR imaging quality and on-device inference metrics.

4.2.4 Impact by high-frequency noise. The system utilizes acoustic signals injected into the gyroscope to control lens jitter, so we assess the impact of surrounding high-frequency noise on our system's imaging performance. In our experimental setup, an additional smartphone (Xiaomi 11) is used to play an interfering 18-22 kHz sweep signal at 100% volume, which is placed 10 cm away from the test smartphone. As shown in Tab. 7, although high-frequency noise does affect lens control and gyroscope signals, the impact on SR performance is minimal due to the severe attenuation of the ambient high-frequency noise signal.

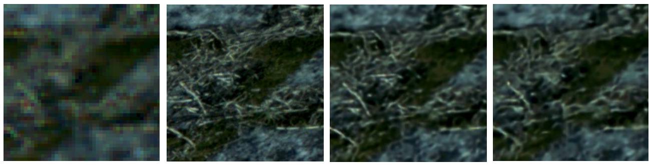
4.3 Night shooting mode

Our method adapts to the vast majority of handheld scenarios. Under sufficient lighting conditions, only particularly significant hand shaking will affect the performance of SR imaging. We determine the extent of hand movement when the user holds the smartphone by examining the amplitude of the optical flow and the readings from the gyroscope. If the movement is too large, we recommend that the user retake the photo.

Under low light conditions, longer exposure times are required for photography, necessitating slower lens movement to minimize the impact of roll shuttering. Additionally, users should ensure the stability of their phones during long exposures. We recommend using a phone holder/tripod for nighttime photography. Fig. 13 demonstrates the results of our system under the night shooting mode. Our setup involves fixing a Xiaomi 11 Pro to a tripod and playing 19.60 kHz acoustic signals, causing the lens to move slowly. The entire shooting process took around 5 s, and the 16× SR imaging results remain exceptional.

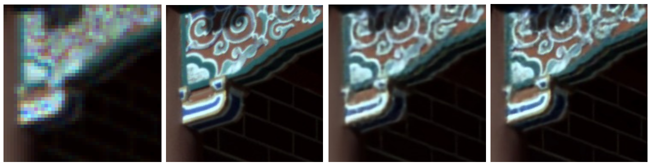
4.4 JPG-format based 16× SR

Our system is not only applicable to SR imaging in RAW shooting mode but also suitable for compressed formats such as PNG and JPG. Without altering the design of the multi-modal optical flow estimation model, we modify the encoder parameters to perform super-resolution imaging on the JPG image input. Furthermore, we assess the performance of our JPG-format based 16×SR system, and the results are shown in Fig. 14. We find that the quality of SR imaging based on the JPG format exhibits somewhat distorted details compared to the RAW format. The on-device inference metrics for the JPG format are: latency of 1.54s, RAM usage of 507.6MB, and energy consumption of 11.43J. These metrics do not present any



(a) LR (b) HR (c) Ours (d) DBSR

Figure 13: Night shot SR performance comparison



(a) LR (b) HR (c) JPG (d) RAW

Figure 14: Results of our M^{3}Cam trained with JPG images

Metrics	PSNR↑	SSIM↑	LPIPS↓	Para. # (10^6)↓	Latency (s) ↓	RAM (MB) ↓	onnx. (MB)↓	Frame #↓	Power (J)↓
EDSR[27]	20.83	0.3865	0.4633	43.08	19.37	1032.3	164	1	91.75
ESRGAN[47]	28.22	0.6782	0.2704	16.69	6.65	812.34	63.8	1	34.26
SwinIR[25]	30.76	0.7599	0.2895	28.01	6.41	483.1	51.6	1	34.87
TR-MISR[2]	29.32	0.7112	0.2479	**0.386**	6.94	572.2	**1.53**	12	31.74
Ours	**36.11**	**0.8777**	**0.1230**	**2.24**	**1.54**	**507.6**	**9.69**	**4**	**11.43**

Table 8: Comparative analysis of 16× SR imaging quality and on-device inference for various JPG-format SR systems

advantage over the RAW format. Therefore, using the RAW format as the input for the SR network yields better imaging results.

We also compare the JPG-format based M^{3}Cam with the SOTA JPG-format based SR system, such as EDSR [27], ESRGAN [47], SwinIR [25], and TR-MISR [2]. The JPG-format based SFSR systems perform poorly in 16× SR tasks due to the fact that all additional pixel information can only be predicted by empirical data. Additionally, their SR system models require a large amount of computational resources, increased latency, and consume considerable energy. Using SFSR methods on mobile devices is not advisable. JPG-format based M^{3}Cam performs better than other JPG-based MFSR systems. Thanks to our multimodal optical flow estimation algorithms, whether we utilize RAW or JPG input images, we can efficiently improve the accuracy of optical flow estimation between the reference frame and offset frames, thereby achieving high-quality 16× SR imaging.

4.5 User Study

In this section, we evaluate the usability of M^{3}Cam. We invited 10 participants to take 10 photos each. We used a standard methodology based on the System Usability Scale (SUS) [17], designing seven questions that participants answered using five options ranging from 'strongly agree (2)' to 'strongly disagree (-2)'. We provided six mainstream SR methods for SR imaging quality comparison with our system, including ESRGAN, BSRT, DBSR, EBSR, BIPNet, and BurstFormer.

We prepared a questionnaire for participants to fill in after using our system. The questionnaire is as follows. **Q1:** I think the experience of handheld shooting with this system is the same as handheld shooting with a normal mobile phone camera app. **Q2:** I think that the high-frequency sound signal during shooting does not cause auditory discomfort. **Q3:** I think the super-resolution image output from the system is clear in detail. **Q4:** I think the quality of the super-resolution image output by the system is better than other methods. **Q5:** I think the time delay in outputting super-resolution images from the system is negligible. **Q6:** I think the system is easy to use. **Q7:** I think I would like to use the system on a regular basis.

The questionnaire results in Tab. 9 showed that most participants expressed a positive view of the system, were willing to use our proposed M^{3}Cam for shooting, and emphasized the ease of use and the quality of SR imaging provided by our system.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Strongly Disagree (-2)	0	1	0	0	1	1	0
Disagree (-1)	1	1	2	2	1	2	1
Not sure (0)	2	3	2	3	1	1	1
Agree (1)	2	2	3	4	5	4	6
Strongly Agree (2)	5	3	3	1	2	2	2
Average Rating	1.1	0.5	0.7	0.4	0.6	0.4	0.9

Table 9: Analysis of questionnaire responses from user study

5 DISCUSSION

Shooting Moving Objects. Considering the brief exposure time of cameras under normal lighting conditions (about 0.03s per frame), slower-moving objects can be assumed stationary for a short period of time. For fast-moving objects, we differentiate them from stationary ones by their distinct optical flow patterns. These moving objects can be masked and processed using SFSR imaging method, which is a common technique for handling moving subjects in MFSR systems [54].

Recording Audio During Imaging. The acoustic injection method uses high-frequency acoustic signals, which are inaudible to humans. Furthermore, these acoustic injection signals can be removed by applying low-pass filtering to the recorded microphone data. We conducted an actual test. While playing high-frequency signals of 18-22KHz on a mobile phone, we recorded the sound using the built-in microphone. The results showed that there was no interference from the high-frequency signals in the recorded audio, and there was almost no low-frequency leakage.

Smartphones from other brands: We tested various smartphone brands, including Xiaomi, Huawei, Oppo, Google Pixel, and Samsung, using an external speaker. All of these devices demonstrated the capability to influence the OIS by inducing regular lens movement through acoustic injection. However, due to the high power of the built-in speakers and the high sensitivity of the IMU's mass, Xiaomi phones exhibited the best performance in affecting the OIS lens motion via their built-in speakers. Consequently, our study exclusively utilized various models of Xiaomi-branded phones. It is important to highlight that our work primarily focuses on incorporating additional modalities related to lens motion in order to develop a lightweight, high-quality SR system for mobile devices. We encourage more smartphone manufacturers to provide direct access to the control interface of the OIS module. Integrating this with our proposed M^{3}Cam would enable the deployment of lightweight 16×SR capabilities across a wider range of devices.

Effects of lens distortion on optical flow estimation: Lens distortions are caused by the non-uniformity of the lens, which typically occurs at the edges of the lens. In our system, we achieve multimodal optical flow estimation by incorporating small movements of the lens into the OIS module. This lens movement is so slight that it results in an optical flow of less than 12 pixels, which is not enough to move the imaging source from the lens distortion region to the undistorted region. In summary, the lens distortion does not affect the optical flow estimation and therefore has no effect on the SR imaging system.

Online learning: We train the SR model on the server, and the deployment on mobile devices does not affect the system's SR inference. Meanwhile, we can continue to collect images and IMU signals from different scenes during the shooting process to expand the dataset. We can further refine and optimize our SR model through techniques such as online learning, thereby adapting it to more shooting scenarios and mobile camera devices. We will explore this in future research.

6 RELATED WORK

6.1 Super resolution in mobile cameras

Existing SR features in mobile phones are generally categorized into two types. The first is zoom-based SR, exemplified by Huawei P40 Pro+ [37], which relies on physical optical zoom. The second type is digital SR, used in applications like Remini [42] and Adobe Lightroom [26], which upload LR images to the cloud for complex SR imaging calculations. However, cloud-based processing can lead to concerns about potential information leakage. In contrast, our proposed M^{3}Cam system is a lightweight and on-device SR solution capable of achieving 16× high-quality SR imaging with real-time inference, eliminating the need for cloud processing.

6.2 Visual Optical Flow

Optical flow estimation is a core technique for many computer vision tasks. Nowadays, numerous visual network frameworks for optical flow have been proposed and applied, treating it as an energy minimization problem that balances a data term against a regularization term. FlowNet [9, 21] is the first CNN for optical flow estimation and is trained in a supervised manner. PWC-Net [43] employs a learned flow field to deform one image before correlation, enhancing its capacity to capture intricate motion details. An asymmetric occlusion-aware feature matching module is proposed in MaskFlowNet [55], which learns to filter occluded regions after feature warping. RAFT [44] defines a Recurrent All-Pairs Field Transforms architecture to combine per-pixel feature extraction, multi-scale correlation volumes, and recurrent updates. Unlike existing optical flow estimation methods, we propose a novel multimodal optical flow estimation network that leverages lens motion information from the OIS module. This approach enables a lightweight network design while achieving sub-pixel level accuracy in optical flow estimation, greatly facilitating subsequent SR synthesis tasks.

6.3 Multi-frame Super Resolution

MFSR system enhances image resolution by combining information from a series of low-resolution frames, exploiting subtle variations between frames to generate a high-resolution output. Handheld [50] and OISSR [39] have demonstrated the possibility of SR reconstruction of multi-frame images by exploiting small offsets generated by hand tremors or acoustic injections during smartphone photography. NeuriCam [46] enhances low-resolution grey-scale video with high-resolution RGB keyframes but is limited to 4× SR, falling short of the 16× SR achieved by SOTA methods. Moreover, existing methods [39, 50] fail to achieve the deployment of the complete system on mobile devices, and server-based schemes [46] inevitably raise privacy concerns among users. DBSR [4, 5] leverages pixel-wise optical flow alignment and attention-based fusion to combine information from multiple LR RAW images. BIPNet [10] utilizes effective pseudo-burst features, edge-boosting burst alignment, and multi-stage resolution enhancement to realize burst image restoration. EBSR [33] presents a novel multi-frame SR architecture that combines Feature Enhanced Pyramid Cascading, Cross Non-Local Fusion, Long Range Concatenation Network, and cascading residual pathways. Furthermore, BSRT [32] employs a pyramid flow-guided deformable convolution network to tackle misalignment and consolidate texture information across multiple frames. Burstormer [11] is a novel transformer-based architecture for burst image restoration and enhancement, achieving SOTA in RAW burst SR, denoising, and low-light enhancement. However, the high computational demands of DNN-based MFSR systems hinder real-time SR inference or mobile deployment. Unlike all the above works, our work cleverly leverages a multimodal optical flow estimation method combined with a swim transformer-based merging network to achieve 16× SR imaging with a lightweight design and high imaging quality.

7 CONCLUSION

We present a lightweight, high-performance 16-fold SR system specifically designed for mobile cameras supporting OIS. Utilizing our proposed multimodal optical flow estimation as its core, the SR system delivers exceptional inference efficiency with minimal resource requirements, enabling on-device SR imaging. When users tap a small area on the screen to zoom in and view details after capturing a scene, our system can instantly provide 16× SR imaging for that area, offering higher-quality photography for an enhanced user experience.

Acknowledgments

We thank the anonymous reviewers for their constructive comments. This work is supported in part by NSFC (No. 61936015, 62072306, 62341201, 62122095, 62432004 and 62072472), Shanghai Key Laboratory of Trusted Data Circulation, Governance and Web3, and a grant from the Guoqiang Institute, Tsinghua University.

References

[1] ONNX AI. 2023. https://onnx.ai/

[2] Tai An, Xin Zhang, Chunlei Huo, Bin Xue, Lingfeng Wang, and Chunhong Pan. 2022. TR-MISR: Multiimage super-resolution based on feature fusion with transformers. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), 1373–1388.

[3] antutu. 2023. https://www.antutu.com/en/doc/index.htm

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. 2021. Deep burst super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9209–9218.

[5] Goutam Bhat, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. 2021. Deep reparametrization of multi-frame super-resolution and denoising. In

Proceedings of the IEEE/CVF International Conference on Computer Vision. 2460–2470.

[6] Brent Cardani. 2006. Optical image stabilization for digital cameras. *IEEE Control Systems Magazine* 26, 2 (2006), 21–22.

[7] Ricardo Omar Chavez-Garcia and Olivier Aycard. 2016. Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking. *IEEE Transactions on Intelligent Transportation Systems* 17, 2 (2016), 525–534. https://doi.org/10.1109/TITS.2015.2479925

[8] Rong Chen, Xiao Tang, Yuxuan Zhao, Zeyu Shen, Meng Zhang, Yusheng Shen, Tiantian Li, Casper Ho Yin Chung, Lijuan Zhang, Ji Wang, et al. 2023. Single-frame deep-learning super-resolution microscopy for intracellular dynamics imaging. *Nature Communications* 14, 1 (2023), 2854.

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision.* 2758–2766.

[10] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2022. Burst image restoration and enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5759–5768.

[11] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2023. Burstormer: Burst image restoration and enhancement transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 5703–5712.

[12] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2023. Burstormer: Burst Image Restoration and Enhancement Transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 5703–5712. https://doi.org/10.1109/CVPR52729.2023.00552

[13] Ming Gao, Feng Lin, Weiye Xu, Muertikepu Nuermaimaiti, Jinsong Han, Wenyao Xu, and Kui Ren. 2020. Deaf-aid: mobile IoT communication exploiting stealthy speaker-to-gyroscope channel. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking.* 1–13.

[14] Google. 2017. Battery Historian. https://github.com/google/battery-historian.

[15] Google. 2023. Inspect CPU activity with CPU Profiler. https://developer.android.com/studio/profile/cpu-profiler.

[16] Google. 2024. Inspect your app's memory usage with Memory Profiler. https://developer.android.com/studio/profile/memory-profiler.

[17] Digital gov. [n. d.]. System Usability Scale (SUS). https://www.usability.gov/how-to-andtools/methods/system-usability-scale.html.

[18] H.W. Haussecker and D.J. Fleet. 2001. Computing optical flow with physical models of brightness variation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 6 (2001), 661–673. https://doi.org/10.1109/34.927465

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[20] Dominik Honegger, Lorenz Meier, Petri Tanskanen, and Marc Pollefeys. 2013. An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. In *2013 IEEE International Conference on Robotics and Automation.* 1736–1741. https://doi.org/10.1109/ICRA.2013.6630805

[21] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. 2018. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 8981–8989.

[22] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2462–2470.

[23] Hakki Can Karaimer and Michael S Brown. 2016. A software platform for manipulating the camera imaging pipeline. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14.* Springer, 429–444.

[24] Bruno Lecouat, Jean Ponce, and Julien Mairal. 2021. Lucas-kanade reloaded: End-to-end super-resolution from raw image bursts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2370–2379.

[25] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision.* 1833–1844.

[26] Adobe Lightroom. [n. d.]. https://lightroom.adobe.com/

[27] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 136–144.

[28] Haisong Liu, Tao Lu, Yihui Xu, Jia Liu, Wenjie Li, and Lijun Chen. 2022. CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 5791–5801.

[29] Jie Liu, Jie Tang, and Gangshan Wu. 2020. Residual feature distillation network for lightweight image super-resolution. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16.* Springer, 41–55.

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision.* 10012–10022.

[31] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. 2022. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 457–466.

[32] Ziwei Luo, Youwei Li, Shen Cheng, Lei Yu, Qi Wu, Zhihong Wen, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. 2022. BSRT: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 998–1008.

[33] Ziwei Luo, Lei Yu, Xuan Mo, Youwei Li, Lanpeng Jia, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. 2021. Ebsr: Feature enhanced burst super-resolution with deformable alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 471–478.

[34] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

[35] Thomas Maschke. 2013. *Digitale kameratechnik: technik digitaler kameras in theorie und praxis.* Springer-Verlag.

[36] MATLAB. 2023. https://ww2.mathworks.cn/help/images/ref/raw2rgb.html

[37] T Mobile. [n. d.]. Huawei P40 Pro Plus review. https://www.techradar.com/reviews/huawei-p40-pro-plus

[38] Hao Pan, Feitong Tan, Yi-Chao Chen, Gaoang Huang, Qingyang Li, Wenhao Li, Guangtao Xue, Lili Qiu, and Xiaoyu Ji. 2022. DoCam: depth sensing with an optical image stabilization supported RGB camera. In *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking.* 405–418.

[39] Hao Pan, Feitong Tan, Wenhao Li, Yi-Chao Chen, and Guangtao Xue. 2022. OISSR: Optical Image Stabilization Based Super Resolution on Smartphone Cameras. In *Proceedings of the 30th ACM International Conference on Multimedia.* 2978–2986.

[40] Anurag Ranjan and Michael J. Black. 2017. Optical Flow Estimation Using a Spatial Pyramid Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[41] Anurag Ranjan and Michael J Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 4161–4170.

[42] Remini. [n. d.]. Remini. https://remini.ai/

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 8934–8943.

[44] Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision.* Springer, 402–419.

[45] Tim Van Erven and Peter Harremos. 2014. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* 60, 7 (2014), 3797–3820.

[46] Bandhav Veluri, Collin Pernu, Ali Saffari, Joshua Smith, Michael Taylor, and Shyamnath Gollakota. 2023. *NeuriCam: Key-Frame Video Super-Resolution and Colorization for IoT Cameras.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3570361.3592523

[47] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops.* 0–0.

[48] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[49] Stephen T Welstead. 1999. *Fractal and wavelet image compression techniques.* Vol. 40. Spie Press.

[50] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. 2019. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (ToG)* 38, 4 (2019), 1–18.

[51] Li Xi, Liu Guosui, and Jinlin Ni. 1999. Autofocusing of ISAR images based on entropy minimization. *IEEE Trans. Aerospace Electron. Systems* 35, 4 (1999), 1240–1252.

[52] Gengshan Yang and Deva Ramanan. 2020. Upgrading Optical Flow to 3D Scene Flow Through Optical Expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 586–595.

[54] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. 2021. Aligned structured sparsity learning for efficient image super-resolution. *Advances in Neural Information Processing Systems* 34 (2021), 2695–2706.

[55] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. 2020. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 6278–6287.