EI-Nexus: Towards Direct and Flexible Inter-Modality Local Feature Extraction and Matching for Event-Image Data (Appendix)

A. Maths

A.1. Repeatability

Given two keypoint sets $E_p = \{(p_i^{E_p}, d_i^{E_p})\}$, $I_p = \{(p_i^{I_p}, d_i^{I_p})\}$ and the homography $\mathbf{H}$ between two imaging plane, only those which can find a corresponding point within a spatial distance threshold of ϵ in another set of keypoints after been warped with $\mathbf{H}$ are treated as valid keypoints. In practice, the $\mathbf{H}$ is an identity matrix when evaluating on events and the corresponding image at the same timestamp. After that, two valid keypoint sets E_{valid} and I_{valid} are filtered out, and $\{(p_i^{E_{valid}}, d_i^{E_{valid}}), (p_i^{I_{valid}}, d_i^{I_{valid}})\}$ are the corresponding valid keypoints. Then, the *Repeatability* is computed as following:

$$\textit{Repeatability} = \frac{|E_{valid}| + |I_{valid}|}{|E_p| + |I_p|}. \tag{1}$$

A.2. VDD and VDA

Given the valid keypoint sets E_{valid} and I_{valid}, the *valid descriptor distance (VDD)* and *valid distance angle (VDA)* are computed as following:

$$\textit{VDD} = \frac{1}{N} \sum_{i=1}^{N} \| d_i^{E_{valid}} - d_i^{I_{valid}} \|_2, \tag{2}$$

$$\textit{VDA} = \frac{1}{N} \sum_{i=1}^{N} \arccos(\| d_i^{E_{valid}} - d_i^{I_{valid}} \|_2). \tag{3}$$

A.3. RPE Ratio and RPE AUC

Given two matched keypoint sets, the essential matrix $\mathbf{E}$ is firstly estimated by using *cv.findEssentialMat()* with RANSAC, then the estimated rotation $\hat{\mathbf{R}}$ and translation $\hat{\mathbf{t}}$ are then recovered. When a list of relative pose estimation results $\{\hat{\mathbf{R}}, \hat{\mathbf{t}}\}$ with corresponding ground truth $\{\mathbf{R}^{gt}, \mathbf{t}^{gt}\}$, the error of $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$ are defined as the angular error between the estimation and the ground truth:

Algorithm 1 RPE AUC Pseudocode, Numpy-like

```
def compute_auc(errors, threshold):
   errors = errors[np.isfinite(errors)]

   sort_idx = np.argsort(errors)
   errors = np.array(errors.copy())[sort_idx]
   recall = (np.arange(len(errors)) + 1) / len(
        errors)
   errors = np.r_[0.0, errors]
   recall = np.r_[0.0, recall]

   last_index = np.searchsorted(errors, threshold)
   rec = np.r_[recall[:last_index], recall[
        last_index - 1]]
   err = np.r_[errors[:last_index], threshold]
   auc = np.trapz(rec, x=err) / threshold
   return auc
```

$$\begin{aligned} \mathbf{R}^{err} &= \frac{tr\left(\hat{\mathbf{R}}^{\top}\mathbf{R}^{gt}\right) - 1}{2}, \\ \mathbf{t}^{err} &= \frac{\hat{\mathbf{t}} - \mathbf{t}^{gt}}{\|\hat{\mathbf{t}}\| \cdot \|\mathbf{t}^{gt}\|}. \end{aligned} \tag{4}$$

The pose error err_i of the current estimation is defined as the maximum error of $\mathbf{R}_i^{err}$ and $\mathbf{t}_i^{err}$. The *RPE Ratio* is then computed:

$$\textit{Ratio} = \frac{|\{err_i \le \epsilon\}|}{|\{err_i\}|}, \tag{5}$$

where ϵ is the specified threshold of angle. For RPE AUC, the area under curve (AUC) is calculated following SiLK [5]. Given a threshold ϵ and the estimation errors $\{err_i\}$, the calculation code is described in Algorithm 1.

A.4. Groundtruth assignment

Given two keypoint sets E_p and I_p, corresponding depth map d^E and d^I, camera matrix $\mathbf{K}^E$ and $\mathbf{K}^I$, and current pose $\mathbf{R}^E, \mathbf{t}^E, \mathbf{R}^I, \mathbf{t}^I$ of event view and image view, the relative poses $\mathbf{T}^{E\rightarrow I}$ and $\mathbf{T}^{I\rightarrow E}$ are computed first. Then, the

Image Extractor	Method	MMA		MR	HE Inlier	HE Ratio			HE AUC		
		$\epsilon=1$	$\epsilon=3$			$\epsilon=3$	$\epsilon=5$	$\epsilon=10$	$\epsilon=3$	$\epsilon=5$	$\epsilon=10$
SuperPoint	E2VID	0.248	0.571	0.448	0.503	0.560	0.772	0.939	**24.17**	41.53	64.55
	HyperE2VID	0.209	0.511	0.395	0.468	0.434	0.691	0.898	16.22	33.62	59.01
	Ours (SuperPoint)	**0.249**	**0.660**	**0.546**	**0.571**	**0.585**	**0.843**	**0.959**	21.00	**42.90**	**67.77**
SiLK	E2VID	0.145	0.303	**0.331**	0.279	0.626	0.772	0.838	29.32	46.27	64.09
	HyperE2VID	0.097	0.217	0.317	0.201	0.444	0.611	0.762	18.58	32.88	51.89
	Ours (SiLK)	**0.267**	**0.485**	0.258	**0.456**	**0.691**	**0.883**	**0.944**	**32.23**	**51.44**	**72.16**

Table 1. **MMA, MR and HE results on EC-RPE set.** MNN is employed for feature matching.

keypoints of one view are projected into the other view:

$$\begin{aligned}\mathbf{p}_i^{E\to I} &= \mathbf{K}^I \frac{1}{z} \mathbf{T}^{E\to I} d_i^E \left(\mathbf{K}^E\right)^{-1} \left[\mathbf{p}_i^E, 1\right]^\top, \\ \mathbf{p}_j^{I\to E} &= \mathbf{K}^E \frac{1}{z} \mathbf{T}^{I\to E} d_j^I \left(\mathbf{K}^I\right)^{-1} \left[\mathbf{p}_j^I, 1\right]^\top,\end{aligned} \tag{6}$$

where z is the normalization term to normalize the depth of the points into unit length. Then the re-projection distance matrix $\mathbf{D}^{M\times N}$ is computed:

$$\mathbf{D}_{ij} = \max\left(\|\mathbf{p}_i^{E\to I} - \mathbf{p}_j^I\|_2^2, \|\mathbf{p}_i^E - \mathbf{p}_j^{I\to E}\|_2^2\right). \tag{7}$$

According to $D^{M\times N}$, the ground-truth assignment $\mathbf{P}^{M\times N}$ is then calculated. Elements $\mathbf{P}_{ij}$ are marked as positive, only if they have the minimum distance in both i-th row and j-th column, and the distance $\mathbf{D}_{ij}$ smaller than a threshold ϵ_p^2:

$$\mathbf{P}_{ij} = \begin{cases} 1, \mathbf{D}_{ij} \le \mathbf{D}_{:j}, \mathbf{D}_{ij} \le \mathbf{D}_{i:}, \mathbf{D}_{ij} < \varepsilon_p^2; \\ 0, others. \end{cases} \tag{8}$$

Finally, the ground-truth matches M^{gt} are obtained through selecting all the (i, j) pairs that have $\mathbf{P}_{ij}$=1.

B. More Implementation Details

B.1. Implementation with SuperPoint

We utilize the SuperPoint architecture and the pre-trained model from the official LightGlue training repository [6], which consists of $1.30M$ parameters in total. The CNN-based backbone encodes the grayscale image $I^{H\times W}$ into latent feature $I_f^{\frac{H}{8}\times\frac{W}{8}\times 128}$. Score head and descriptor head separately predict a score map $I_{score}^{\frac{H}{8}\times\frac{W}{8}\times 65}$ and a descriptor map $I_{desc}^{\frac{H}{8}\times\frac{W}{8}\times 256}$. Then the dustbin dimension of I_{score} is removed and $I_{score}^{\frac{H}{8}\times\frac{W}{8}\times 64}$ is reshaped into $I_{score}^{H\times W\times 1}$ through pixel shuffle. After that, the keypoint extraction procedure from SiLK [5] is employed, which produces a set of keypoint positions p_i. The d_i of the keypoint in p_i is then extracted from the normalized I_{desc} through bilinear sampling.

When employing the event extractor $\mathcal{E}_E$ corresponding to SuperPoint, the dimensions of the latent feature, score map, and descriptor map are supposed to be the same as SuperPoint, due to the implementation of the proposed local feature distillation. In practice, we construct a VGG-like architecture as $\mathcal{E}_E$ that has $1.31M$ parameters. During training, we perform the cosine learning schedule with an initial learning rate of 1×10^{-3} for 50 epochs, and the batch size is set to 8.

B.2. Implementation with SiLK

We perform the official SiLK model that uses no max-pooling and consists of $1.57M$ parameters. In this case, it generates $I_f^{H\times H\times 128}$, $I_{score}^{H\times W\times 1}$, $I_{desc}^{H\times W\times 128}$ in full resolution. Therefore, we do not use max-pooling in our VGG-like $\mathcal{E}_E$ with $1.10M$ parameters when training with SiLK. The learning hyper-parameters are set the same as used in the SuperPoint implementation.

B.3. Mutual Nearest Neighbor

Give two keypoint sets E_p=$\{p_i^E, d_i^E\}$ and I_p=$\{p_i^I, d_i^I\}$, a similarity matrix $\mathbf{S}^{M\times N}$ is firstly computed:

$$\mathbf{S}_{ij} = \left(d_i^E\right)^\top d_j^I. \tag{9}$$

The estimated assignment $\hat{\mathbf{P}}$ is obtained by applying a softmax operation followed by a logarithm on each axis:

$$\hat{\mathbf{P}} = \log\left(\frac{\exp\left(\mathbf{S}_{ij}\right)}{\sum\exp\left(\mathbf{S}_{:j}\right)}\right) + \log\left(\frac{\exp\left(\mathbf{S}_{ij}\right)}{\sum\exp\left(\mathbf{S}_{i:}\right)}\right). \tag{10}$$

The final predicted matches $\hat{\text{M}}$ are obtained through filtering all the possible (i, j) pairs where $\hat{\mathbf{P}}_{ij}$ have the largest value among the i-th row and the j-th column.

B.4. LightGlue

LightGlue is a representative Context Aggregation (CA) method that uses attention techniques to aggregate information within the keypoint set and between keypoint sets. We follow the official implementation of LightGlue, but ignore the point pruning and early stop procedure, which are designed to boost inference speed and do not affect the matching performance. During training, we utilize a cosine schedule with an initial learning rate of 1×10^{-4} for 50 epochs. The batch size is set to 8.

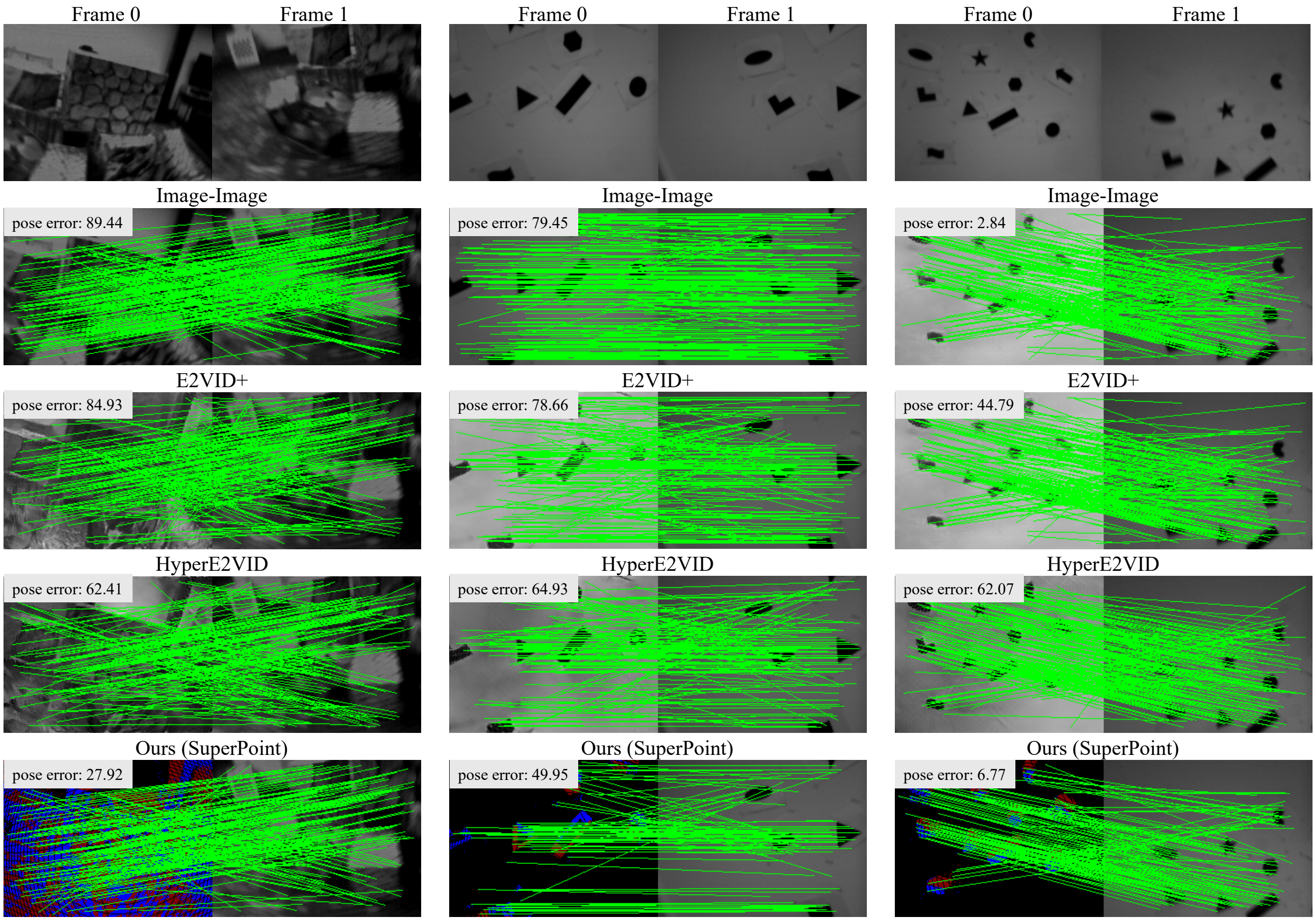

Figure 1. **Matching results on EC-RPE set for relative pose estimation.** Features from the image are extracted by SuperPoint.

C. Additional Experiments

C.1. More Cross-modal Keypoint Similarity Results

Mean Matching Accuracy and Matching Rate. Following SiLK [5], we further evaluate the cross-modal keypoint similarity with *mean matching accuracy (MMA)*. Given the events and image at the same timestamp, the MMA measures the accuracy of the valid matching pairs by applying an MNN for feature matching. We set two thresholds ϵ=1 and ϵ=3 to obtain valid checks. In addition, the *matching rate (MR)* which calculates the ratio of the matched pairs is also evaluated for a more comprehensive comparison. The MMA and MR results are shown in Table 1. Our framework surpasses the explicit transform methods by a large margin in most situations, showing the best keypoint similarity among all methods.

Homography Estimation. Since the events and image are at the same timestamp when evaluating the cross-modal keypoint similarity, the homography between two imaging planes is an identity transform. We follow previous feature matching methods [2, 5, 7] to construct a Homography Estimation (HE) task for event-image feature matching. The matched keypoint pairs are used for estimating a homography $\hat{\mathbf{H}}$ using *cv.findHomography()* with RANSAC. The corner error between the images warped with the estimated homography $\hat{\mathbf{H}}$ and the ground-truth homography $\mathbf{H}^{gt}$=$\mathbf{I}$ is then computed for correctness identification. Lastly, the *HE Inlier* ratio from RANSAC, the *HE Ratio* and *HE AUC* under different thresholds are computed. As shown in Table 1, our method achieves the most accurate estimation result, emphasizing the superiority of the direct inter-modality feature matching proposed by EI-Nexus.

C.2. More Relative Pose Estimation Results

We further show the inter-modality RPE results on the EC-RPE set. As shown in Fig. 1, the RPE results of EI-Nexus are much better than event-to-video methods, and even better than image-image results for some instances. It indicates that the image local feature extraction methods that trained on a specific dataset, could not always perform well in new scenes, and no suitable fine-tuning approach for now is presented for image-based local extraction. In contrast, our proposed framework can achieve superior and stable performance through the use of the simple yet effective

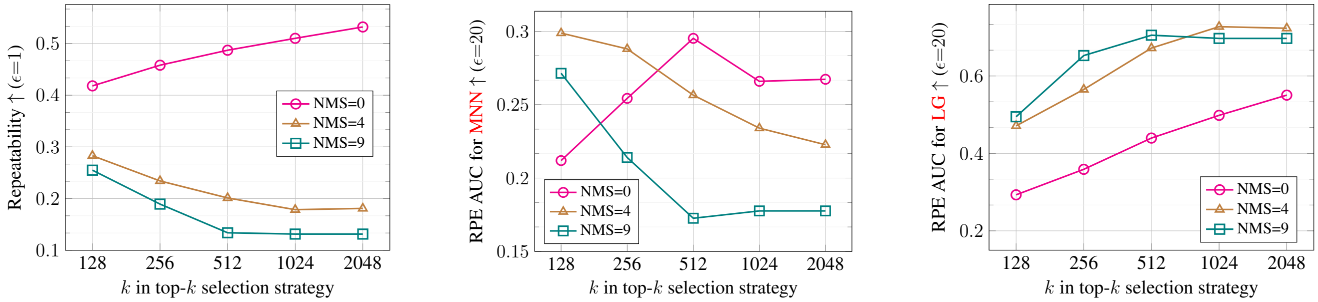

Figure 2. **Comparison of different post-processing parameters of keypoint extraction and different matchers.** The keypoint similarity and RPE metrics are evaluated and compared across a variety of different test scenarios.

local feature distillation (LFD) approach. This highlights the broad applicability and ease of implementation of our method.

C.3. Extraction Post-processing Parameters and Properties of Matchers.

The hyper-parameters governing the keypoint extraction process exert a decisive influence on the quality of the final keypoints. Since the border removal is to prevent keypoint selection from fake edge information, we do not consider that and investigate the influence of the NMS radius and top-k instead.

As presented in Fig. 2, it is observed that the trends for the settings of NMS=4 and NMS=9 exhibit similar patterns. When enlarging k, the *Repeatability* decreases, meaning the quality of the keypoint set declines. Concurrently, the MNN exhibits worse RPE performance when k is larger, while the LG behaves in the opposite manner. We also notice that the performance with NMS=9 does not change when the k value exceeds 1024, as the top-k keypoint selection is unable to detect additional keypoints when the score map becomes too sparse after applying a large NMS radius. In addition, the performance using an NMS radius of 4 is usually better than 9, except for the *RPE AUC* when utilizing the LG approach. This is because the usage of the position encoding guides the LG method to focus more on the points that are spread widely.

When NMS is not applied, the extracted keypoints are clustered since the scores around a keypoint are usually close to it. In this case, a region of keypoints will be extracted without NMS, resulting in a higher probability of having an intersection area with another region. However, the matching results are not satisfying when applying MNN because lack of points in different fields of view, unless the extracted points are enough but not too much. In addition, for the LG method, the *RPE AUC* could not achieve good results because of the use of position encoding.

The observations presented above suggest that the post-processing procedure of keypoint extraction is highly important for local feature extraction and downstream feature-matching tasks. In addition, the analyses underscore the importance of choosing appropriate post-processing parameters for specific scenarios and models.

Method	AVG Pose Error↓	RPE Ratio ↑			RPE AUC ↑		
		$\epsilon = 5°$	$\epsilon = 10°$	$\epsilon = 20°$	$\epsilon = 5°$	$\epsilon = 10°$	$\epsilon = 20°$
HyperE2VID	48.46	0.00	0.17	0.21	0.00	7.04	13.54
Ours	**42.32**	**0.04**	**0.26**	**0.34**	**2.74**	**10.06**	**21.03**

Table 2. **Relative pose estimation results on the EVIMO2 dataset.**

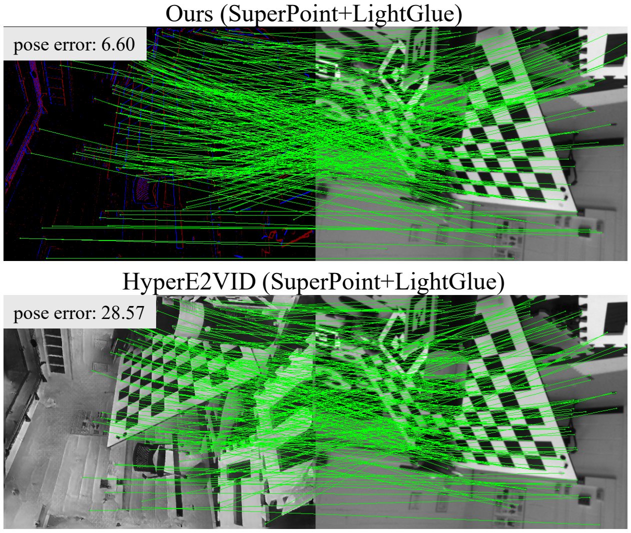
Ours (SuperPoint+LightGlue)
pose error: 6.60
HyperE2VID (SuperPoint+LightGlue)
pose error: 28.57

Figure 3. **Matching results on EVIMO2 dataset.** Features are extracted by SuperPoint and matched by LightGlue.

C.4. Results on Depth-aware System

For depth-aware systems [1, 3, 4], in which the event camera and regular camera are deployed separately, the pixel-level correspondence could be obtained by reprojecting the image from the imaging plane of the regular camera into the imaging plane of event camera according to the extrinsic parameters and the depth of each pixel in the original image.

Following this pipeline, we test our framework on the EVIMO2 dataset, which contains an RGB camera with 2080×1552 resolution and two Prophesee cameras with 640×480 resolution. We use the sequences from the *sfm* scenario in our experiments.

Method	Data Processing (ms)	Model	Extractor (ms)	Matcher (ms)
E2VID+	10.9	SP+MNN	18.5	35.2
		SP+LG	17.8	51.7
HyperE2VID	10.8	SiLK+MNN	29.1	37.7
		SiLK+LG	29.3	49.1
Ours	46.0	SP+MNN	19.1	36.5
		SP+LG	19.1	53.5
		SiLK+MNN	29.3	34.5
		SiLK+LG	28.9	54.5

Table 3. **Comparison on inference time.** SP represents SuperPoint [2] and LG represents LightGlue [6].

The event extractor is trained through LFD using the events and reprojected images. Then we train LightGlue as a learnable matcher using the events and original images, given their ground-truth relative poses. For evaluation, the events from the event camera and the original image from the regular camera at the same timestamp are used for matching. The matches are then used to estimate the extrinsic parameter between those two cameras. Results are shown in Table 2 and quantitative results are shown in Fig. 3. In such a depth-aware system, our model still works better than those that apply explicit modality transformation first. It should be emphasized that our method only needs events from a short time interval $[t_j - \Delta t, t_j]$, while the event-to-video methods need long-term previous information of the sequence.

C.5. Inference Time

We give the inference time of the models tested on an NVIDIA A800 GPU. Data processing time for event-to-video methods means the time consumption for reconstructing one frame. Our model converts the voxel grid and computes an event mask during data processing. As shown in Table 3, the entire time consumption of our model lies in the data processing procedure. For the computation cost of the network, the inference time is almost the same as event-to-video methods, as the only difference of the network between ours and events-to-video methods is the input channels of the first CNN layer. In addition, we find that the SiLK costs more computation than SuperPoint as its backbone does not have a pooling operation.

D. Limitations

Despite the impressive performance of EI-Nexus in event-image inter-modality local feature extraction and matching, there remain several limitations that warrant further investigation. First, The event representations explored in this work are the only commonly used methods that convert the event stream into 2D representations, which may not fully capture the spatial-temporal information for inter-modality local feature extraction and matching. Future research could explore more expressive representation approaches, such as learning-based techniques, to improve the robustness of the framework. Second, the intra-modality performance of the learned event extractor is not investigated. Although the whole EI-Nexus framework is designed for inter-modality tasks, a unified framework for both intra-modality and inter-modality local feature extraction and matching is preferred in downstream applications. However, since EI-Nexus provides a simple, direct, and flexible approach to constructing the local feature relationship of different modalities, it could be an indispensable part of such a unified multi-modality framework. Third, the proposed EI-Nexus solution is supposed to train on a specific dataset, which limits its generalization ability. In this case, further research on training on synthetic data is encouraged.

References

[1] Levi Burner, Anton Mitrokhin, Cornelia Fermüller, and Yiannis Aloimonos. EVIMO2: An event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. *arXiv preprint arXiv:2205.03467*, 2022. 4

[2] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 3, 5

[3] Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. VECtor: A versatile event-centric benchmark for multi-sensor SLAM. *IEEE Robotics and Automation Letters*, 2022. 4

[4] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. 4

[5] Pierre Gleize, Weiyao Wang, and Matt Feiszli. SiLK: Simple learned keypoints. In *ICCV*, 2023. 1, 2, 3

[6] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. In *ICCV*, 2023. 2, 5

[7] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 3