A Simple and Effective Point-based Network for Event Camera 6-DOFs Pose Relocalization

Hongwei Ren*, Jiadong Zhu*, Yue Zhou, Haotian Fu, Yulong Huang, Bojun Cheng †
The Hong Kong University of Science and Technology(Guangzhou)
{hren066,jzhu484,yzhou833,hfu373,yhuang496}@connect.hkust-gz.edu.cn, bocheng@hkust-gz.edu.cn

Abstract

Event cameras exhibit remarkable attributes such as high dynamic range, asynchronicity, and low latency, making them highly suitable for vision tasks that involve high-speed motion in challenging lighting conditions. These cameras implicitly capture movement and depth information in events, making them appealing sensors for Camera Pose Relocalization (CPR) tasks. Nevertheless, existing CPR networks based on events neglect the pivotal fine-grained temporal information in events, resulting in unsatisfactory performance. Moreover, the energy-efficient features are further compromised by the use of excessively complex models, hindering efficient deployment on edge devices. In this paper, we introduce PEPNet, a simple and effective point-based network designed to regress six degrees of freedom (6-DOFs) event camera poses. We rethink the relationship between the event camera and CPR tasks, leveraging the raw Point Cloud directly as network input to harness the high-temporal resolution and inherent sparsity of events. PEPNet is adept at abstracting the spatial and implicit temporal features through hierarchical structure and explicit temporal features by Attentive Bi-directional Long Short-Term Memory (A-Bi-LSTM). By employing a carefully crafted lightweight design, PEPNet delivers state-of-the-art (SOTA) performance on both indoor and outdoor datasets with meager computational resources. Specifically, PEPNet attains a significant 38% and 33% performance improvement on the random split IJRR and M3ED datasets, respectively. Moreover, the lightweight design version PEPNet$_{tiny}$ accomplishes results comparable to the SOTA while employing a mere 0.5% of the parameters.

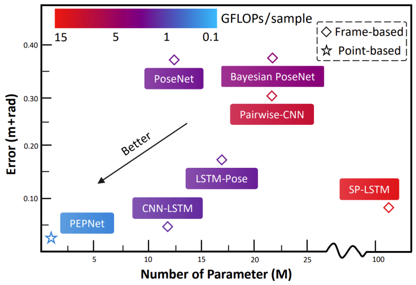

Figure 1. The average results using the random split method benchmarked on the CPR dataset [23]. The vertical axis represents the combined rotational and translational errors (m+rad). PEPNet is the first point-based CPR network for event cameras.

1. Introduction

Event camera is a type of bio-inspired vision sensor that responds to local changes in illumination exceeding a predefined threshold [17]. Differing from conventional frame-based cameras, event cameras independently and asynchronously produce pixel-level events. Notably, event cameras boast an exceptional triad: high dynamic range, low latency, and ultra-high temporal resolution. This unique combination empowers superior performance under challenging light conditions, adeptly capturing the swift scene and rapid motion changes in near-microsecond precision [27]. Additionally, event cameras boast remarkably low power consumption positioning them as a popular choice for many power-constrained devices. Camera Pose Relocalization (CPR) is an emerging application in power-constrained devices and has gained significant attention. It aims to train several scene-specific neural networks to accurately relocalize the camera pose within the original scene used for training. It is extensively employed in numerous applications, including Virtual Reality (VR), Augmented Reality (AR), and robotics [35], all of which are deployed on battery-powered devices and are power-constrained.

CPR tasks using event cameras significantly diverge from their conventional CPR counterpart that employs frame-based cameras, primarily due to the inherent dissimilarity in data output mechanisms between these two camera types. Furthermore, events inherently encompass information regarding object motion and depth changes

*equal contribution. †corresponding author.

across precise temporal and spatial dimensions attributes of paramount significance within the domain of CPR tasks [8, 31]. Regrettably, existing event-based CPR networks often derive from the conventional camera network paradigms and inadequately address the unique attributes of event data. More specifically, events are transformed into various representations such as event images [26], time surfaces [18], and other representations[18], leading to the loss of their fine-grained temporal information. Furthermore, most event-based methods tend to overlook the computational load of the network, only prioritizing elevated accuracy, which contradicts the fundamental design principles of event cameras [9].

A suitable and faithful data representation is crucial for event cloud processing. Point Cloud is a collection of 3D points (x, y, z) that represents the shape and surface of an object or environment commonly used in lidar and depth cameras [10]. The distance (z) is of great meaning to the tasks. As for event camera, by treating each event's temporal information as the third dimension, event inputs (x, y, t) can be regarded as points and aggregated into a pseudo-Point Cloud [28, 29, 32–34, 40]. However, given that the t dimension of Event Cloud is not strictly equivalent to the spatial dimensions (x, y, z), direct transplantation of the Point Cloud network has not yet exhibited a satisfactory performance advantage in processing event data [32, 40].

In this study, we introduce PEPNet, the first point-based end-to-end CPR network designed to harness the attributes of event cameras. A comparison of our performance and method to other frame-based methods is illustrated Fig. 1 and Fig. 2, respectively. Moreover, diverging from other point-based approaches in event data processing [32, 40], PEPNet demonstrates careful attention to detail by systematically assessing the difference between Event Cloud and Point Cloud in its design approach. This approach enables a more precise extraction of spatio-temporal features and facilitates solutions for a spectrum of event-based tasks. Our main contributions are as follows: First, in the preprocessing stage, PEPNet directly processes the raw data obtained from the event cameras, meticulously preserving the fine-grained temporal coordinate and the order inherent in the event data. Second, PEPNet proficiently captures spatial features and **implicit temporal** features through its hierarchical structure with temporal aggregation. Subsequently, the **explicit temporal** feature is processed by the A-Bi-LSTM, thanks to the preservation of the input sequence in previous stages. As such, this architecture is tailored to accommodate the high temporal resolution and sparse characteristics inherent in event cameras. Thirdly, by restricting ourselves to minimal hardware resources and deliberately avoiding heavy computational modules, PEPNet not only attains SOTA results on IJRR [23] and M3ED [4] dataset but also features a lightweight design that can be executed in real-time. We hope such an approach could potentially democratize computer vision technology by making it accessible to a wider range of devices and applications in the community of edge computing.

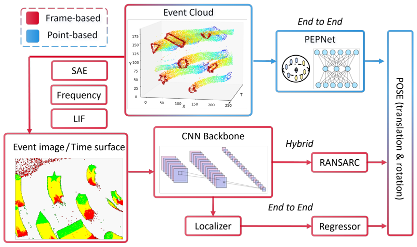

Figure 2. Two different event-based processing methods, frame-based and point-based.

2. Related Work

2.1. Frame-based CPR Learning Methods

Deep learning, crucial for vision tasks like classification and object detection [16], has seen advancements such as PoseNet's innovative transfer learning [14]. Utilizing VGG, ResNet [11, 36], LSTM, and customized loss functions [25, 39, 41], researchers enhanced this approach. Auxiliary Learning methods further improved performance [19, 30, 38], although overfitting remains a challenge. Hybrid pose-based methods, combining learning with traditional pipelines [1, 15], offer promise. DSAC series, for instance, achieve high pose estimation accuracy [2, 3], but come with increased computational costs and latency, especially for edge devices.

2.2. Event-based CPR Learning Methods

Event-based CPR methods often derive from the frame-based CPR network. SP-LSTM [26] employed the stacked spatial LSTM networks to process event images, facilitating a real-time pose estimator. To address the inherent noise in event images, [12] proposed a network structure combining denoise networks, convolutional neural networks, and LSTM, achieving good performance under complex working conditions. In contrast to the aforementioned methods, a novel representation named Reversed Window Entropy Image (RWEI) [18] is introduced, which is based on the widely used event surface [22] and serves as the input to an attention-based DSAC* pipeline [2] to achieve SOTA results. However, the computationally demanding architecture involving representation transformation and hybrid pipeline poses challenges for real-time execution. Additionally, all existing methods ignore the fine-grained temporal feature of the event cameras, and accumulate events into frames for processing, resulting in unsatisfactory performance.

2.3. Point Cloud Network

Point-based methodologies have transformed the direct processing of Point Cloud, with PointNet [28] as a standout example. Taking a step beyond, PointNet++ [29] introduced a Set Abstraction module. While it initially employed a straightforward MLP in the feature extractor, recent advancements have seen the development of more sophisticated feature extractors to enhance Point Cloud processing [5, 21, 42, 44]. When extending these techniques to Event Cloud, Wang et al. [40] addressed the temporal information processing challenge while maintaining representation in both the x and y axes, enabling gesture recognition using PointNet++. Further enhancements came with PAT [43], which incorporated self-attention and Gumbel subset sampling, leading to improved performance in recognition tasks. However, existing point-based models still fall short in performance compared to frame-based methods. This phenomenon can be attributed to the distinctively different characteristics of Point Cloud and Event Cloud. Event Cloud contradicts the permutation and transformation invariance present in Point Cloud due to its temporal nature. Additionally, the Point Cloud network is not equipped to extract explicit temporal features.

3. PEPNet

PEPNet pipeline consists of four essential modules: (1) a preprocessing module for the original Event Cloud, (2) a hierarchical Point Cloud feature extraction structure, (3) an Attentive Bi-directional LSTM, and (4) a 6-DOFs pose regressor, as illustrated in Fig. 3. In the following sections, we will provide detailed descriptions and formulations for each module.

3.1. Event Cloud

To preserve the fine-grained temporal information and original data distribution attributes from the Event Cloud, the 2D-spatial and 1D-temporal event information is constructed into a three-dimensional representation to be processed in Point Cloud. Event Cloud consists of time-series data capturing spatial intensity changes of images in chronological order, and an individual event is denoted as $e_k = (x_k, y_k, t_k, p_k)$, where k is the index representing the k_{th} element in the sequence. Consequently, the set of events within a single sequence ($\mathcal{E}$) in the dataset can be expressed as:

$$\mathcal{E} = \{e_k = (x_k, y_k, t_k, p_k) \mid k = 1, \ldots, n\} \tag{1}$$

For a given pose in the dataset, the ground truth resolution is limited to 5 ms, while the event resolution is 1 μs. Therefore, it is necessary to acquire the events that transpire within the time period we call it sliding window corresponding to the poses, which will serve as the input for the model, as depicted by the following equation:

$$P_i = \{e_{j \to l} \mid t_l - t_j = R\} \quad i = 1, \ldots, M \tag{2}$$

The symbol R represents the time interval of the sliding window, where j and l denote the start and end event index of the sequence, respectively. The variable M represents the number of sliding windows into which the sequence of events $\mathcal{E}$ is divided. Before being fed into the neural network, P_i also needs to undergo sampling and normalization. Sampling is to unify the number of points N as network inputs. We set $N = 1024$ in PEPNet. Additionally, as the spatial coordinates are normalized by the camera's resolution w and h. The normalization process is described by the following equation:

$$PN_i = (\frac{X_i}{w}, \frac{Y_i}{h}, \frac{T_i - t_j}{t_l - t_j}) \tag{3}$$

$$X_i, Y_i, T_i = \{x_j, \ldots, x_l\}, \{y_j, \ldots, y_l\}, \{t_j, \ldots, t_l\} \tag{4}$$

The X, Y is divided by the resolution of the event camera. To normalize T, we subtract the smallest timestamp t_j of the window and divide it by the time difference $t_l - t_j$, where t_l represents the largest timestamp within the window. After pre-processing, Event Cloud is converted into the pseudo-Point Cloud, which comprises explicit spatial information (x, y) and implicit temporal information t.

3.2. Hierarchy Structure

The hierarchy structure is the backbone for processing the pseudo-3D Point Cloud and is composed of four primary modules: grouping and sampling, standardization, feature extractor, and aggregation, as described in the following subsection. To efficiently extract deeper explicit spatial and implicit temporal features, the hierarchical structure is tailored and differs from conventional hierarchical structure in a few ways: First, we no longer force permutation invariance as usually done in mainstream point-based methods [21, 28], as the motion information is inherently related to the sequential order of events. Instead, we **keep the sequence of all events strictly in the same order** as they are generated to preserve the temporal information to be used in the next stage. Second, we replace MaxPooling in aggregation and deploy temporal aggregation which leverages the attention mechanism with softmax, which improves the effective assimilation of temporal information into the resultant feature vectors.

3.2.1 Grouping and Sampling

Aligned with the frame-based design concept, our focus is to capture both local and global information. Local information is acquired by leveraging Farthest Point Sampling

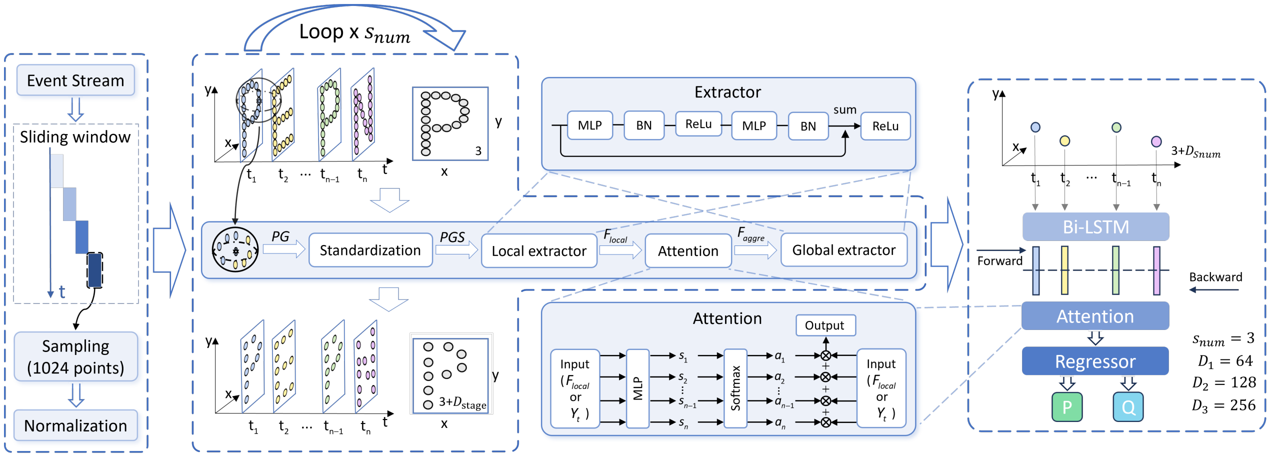

Figure 3. PEPNet overall architecture (the time resolution of $t_1, t_2, ...t_n$ is $1\mu s$). The input Event Cloud undergoes direct handling through a sliding window, sampling, and normalization, eliminating the need for any format conversion. Sequentially, the input passes through S_{num} hierarchy structures for spatial feature abstraction and extraction. It further traverses a bidirectional LSTM for temporal feature extraction, culminating in a regressor responsible for 6-DOFs camera pose relocalization.

(FPS) and K-Nearest Neighbors (KNN), while global information is obtained through a dedicated aggregation module.

$$PS_i = FPS(PN_i) \quad PG_i = KNN(PN_i, PS_i) \tag{5}$$

The input dimension PN_i is $[N, 3+D]$, and the centroid dimension PS_i is $[N^{'}, 3+D]$ and the group dimension PG_i is $[N^{'}, K, 3+2*D]$. K represents the nearest K points of the center point (centroid), D is the feature dimension of the points of the current stage, and 3 is the most original (X, Y, T) coordinate value. Importantly, it should be noted that the ordering of all points in the grouping and sampling process strictly adheres to the timestamp (T), and the dimension $2*D$ of the points in the group is the result of being concatenated to the centroid.

3.2.2 Standardization

Next, each group undergoes a standardization process to ensure consistent variability between points within the group, as illustrated in this formula:

$$PGS_i = \frac{PG_i - PS_i}{Std(PG_i)} \quad Std(PG_i) = \sqrt{\frac{\sum_{j=0}^{3n-1}(g_j - \bar{g})^2}{3n-1}} \tag{6}$$

$$g = [x_0, y_0, t_0, \ldots, x_n, y_n, t_n] \tag{7}$$

Where PG_i and PS_i are the subsets of PG and PS, Std is the standard deviation, the dimension of $Std(PG)$ is M which is consistent with the number of sliding windows, and g is the set of coordinates of all points in the PG_i.

3.2.3 Feature extractor

Following the standardization of PG by dividing the variance by the subtracted mean, the feature extraction is performed using a Multi-Layer Perceptron (MLP) with a residual connection. This process encompasses two steps: local feature extraction and global feature extraction. The feature extractor with a bottleneck can be mathematically represented as:

$$I(x) = f(\text{BN}(\text{MLP}_1(x))) \tag{8}$$

$$O(x) = \text{BN}(\text{MLP}_2(x)) \tag{9}$$

$$Ext(x) = f(x + O(I(x))) \tag{10}$$

BN represents batch normalization layer, while f signifies the nonlinear activation function. Both local feature extraction and global feature extraction maintain identical input and output dimensions. The dimension increase occurs solely when combining the feature dimension D of the current point with the feature dimension D of the centroid during grouping, resulting in a final dimension of $2*D$. The feature extractor takes an input dimension of $[B, N, K, D]$, and following local feature extraction, the dimension remains $[B, N, K, D]$, B represents batch size. We adopt the attention mechanism for aggregation, yielding an aggregated feature dimension of $[B, N, D]$. Subsequently, the aggregated feature map is then processed through the global feature extractor, completing the feature extraction for the current stage.

3.2.4 Temporal Aggregation

Conventional Point Cloud methods favor MaxPooling operations for feature aggregation because it is efficient in extracting the feature from one point among a group of points and discarding the rest. However, MaxPooling involves extracting only the maximum value along each dimension of the temporal axis. It is robust to noise perturbation but also ignores the temporal nuances embedded within the features. Conversely, the integration of attention mechanisms enhances the preservation of those nuanced and useful temporal attributes by aggregating features along the temporal axis through the attention value. To provide a more comprehensive exposition, we employ a direct attention mechanism within the K temporal dimensions to effectively aggregate features as shown in Fig. 3. This mechanism enables the explicit integration of temporal attributes, capitalizing on the inherent strict ordering of the K points. The ensuing formula succinctly elucidates the essence of this attention mechanism:

$$F_{\text{local}} = Ext(x) = (F_{t1}, F_{t2}, \ldots, F_{tk}) \tag{11}$$

$$A = \text{SoftMax}(\text{MLP}(F_{\text{local}})) = (a_{t1}, a_{t2}, \ldots, a_{tk}) \tag{12}$$

$$F_{\text{aggre}} = A \cdot F_{\text{local}} = F_{t1} \cdot a_{t1} + F_{t2} \cdot a_{t2} + \cdots + F_{tk} \cdot a_{tk} \tag{13}$$

Upon the application of the local feature extractor, the ensuing features are denoted as F_{local}, and F_{tk} mean the extracted feature of k_{th} point in a group. The attention mechanism comprises an MLP layer with an input layer dimension of D and an output a_{tk} dimension of 1, along with softmax layers. Subsequently, the attention mechanism computes attention values, represented as A. These attention values are then multiplied with the original features through batch matrix multiplication, resulting in the aggregated feature F_{aggre}.

3.3. A-Bi-LSTM

The temporal features extracted through the hierarchical structure are independent and parallel, lacking recurrent mechanisms within the network. This distinctive attribute, referred to as 'implicit', contrasts with the conventional treatment of temporal information as an indexed process. Consequently, implicit temporal features **inadequately capture the interrelations among events along the timeline**, whereas explicit temporal features assume a pivotal role in facilitating the CPR task. To explicitly capture temporal patterns, we introduce the LSTM network, which has been proven effective in learning temporal dependencies. For optimal network performance, controlled feature dimensionality, and comprehensive capture of bidirectional relationships in pose context, we adopt a bidirectional LSTM network with a lightweight design. The regressor attentively focuses on the output of Bi-LSTM at each timestep and is more inclined towards the start and end features as demonstrated in Fig. 6. The integration of bidirectional connections into the recurrent neural network (RNN) is succinctly presented through the following equation:

$$\mathbf{h}_t = f(\mathbf{W}_h \cdot \mathbf{x}_t + \mathbf{U}_h \cdot \mathbf{h}_{t-1} + \mathbf{b}_h) \tag{14}$$

$$\mathbf{h}'_t = f(\mathbf{W}'_h \cdot \mathbf{x}_t + \mathbf{U}'_h \cdot \mathbf{h}'_{t+1} + \mathbf{b}'_h) \tag{15}$$

$$\mathbf{y}_t = \mathbf{V} \cdot \mathbf{h}_t + \mathbf{b}_y \quad \mathbf{y}'_t = \mathbf{V}' \cdot \mathbf{h}'_t + \mathbf{b}'_y \tag{16}$$

$\mathbf{x}_t$ represents the feature vector at the t-th time step of the input sequence, while $\mathbf{h}_{t-1}$ and $\mathbf{h}'_{t+1}$ correspond to the hidden states of the forward and backward RNN units, respectively, from the previous time step. The matrices $\mathbf{W}_h$, $\mathbf{U}_h$, and $\mathbf{b}_h$ denote the weight matrix and bias vector of the forward RNN unit, while $\mathbf{V}$ and $\mathbf{b}_y$ represent the weight matrix and bias vector of its output layer. Similarly, $\mathbf{W}'_h$, $\mathbf{U}'_h$, and $\mathbf{b}'_h$ are associated with the weight matrix and bias vector of the backward RNN unit, and $\mathbf{V}'$ and $\mathbf{b}'_y$ pertain to the weight matrix and bias vector of its output layer. The activation function, denoted as $f(\cdot)$, can be chosen as sigmoid or tanh or other functions. The final output Y_a is aggregated at each moment using the attention mechanism, and $\oplus$ means concat operation.

$$Y_t = y_t \oplus y'_t \tag{17}$$

$$A = \text{SoftMax}(\text{MLP}(Y_t)) \tag{18}$$

$$Y_a = A \cdot Y_t \tag{19}$$

3.4. Loss Function

A fully connected layer with a hidden layer is employed to address the final 6-DOFs pose regression task. The displacement vector of the regression is denoted as $\hat{p}$ representing the magnitude and direction of movement, while the rotational Euler angles are denoted as $\hat{q}$ indicating the rotational orientation in three-dimensional space.

$$\text{Loss} = \alpha||\hat{p} - p||_2 + \beta||\hat{q} - q||_2 + \lambda \textstyle\sum_{i=0}^{n} w_i^2 \tag{20}$$

p and q represent the ground truth obtained from the dataset, while α, β, and λ serve as weight proportion coefficients. In order to tackle the prominent concern of overfitting, especially in the end-to-end setting, we incorporate the L2 regularization into the loss function. This regularization, implemented as the second paradigm for the network weights w, effectively mitigates overfitting.

3.5. Overall Architecture

Next, we will present the PEPNet pipeline in pseudo-code, utilizing the previously defined variables and formulas as described in Algorithm 1.

4. Experiment

In this section, we present an extensive and in-depth analysis of PEPNet's performance on both indoor and outdoor datasets, encompassing evaluations based on rotational and translational mean squared error (MSE), model parameters, floating-point operations (FLOPs), and inference time. PEPNet's training and testing are performed on a server furnished with an AMD Ryzen 7950X CPU, an RTX GeForce 4090 GPU, and 32GB of memory.

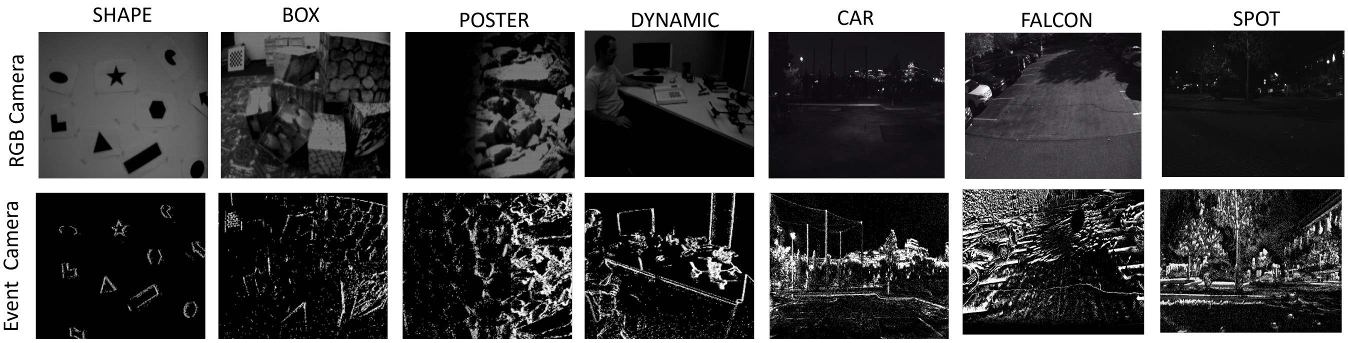

Figure 4. Event-based CPR Dataset visualization.

Algorithm 1 PEPNet pipeline

Input: Raw event stream $\mathcal{E}$

Parameters: $N_p = 1024, R = 1e + 3, S_{\text{num}} = 3, K = 24$

Output: 6-DOFs pose $(\hat{p}, \hat{q})$

```
Preprocessing
for $j$ in len($\mathcal{E}$) do
    $P_i$.append($e_{j\to l}$) ; $j = l$; where $t_l - t_j = R$
    if ($len(P_i) > N_p$): $i = i + 1$;
end for
$PN$ = Normalize(Sampling($P$))
Hierarchy structure
for stage in range($S_{\text{num}}$) do
    Grouping and Sampling($PN$)
    Get $PGS \in [B, N_{\text{stage}}, K, 2 * D_{\text{stage}-1}]$
    Local Extractor($PGS$)
    Get $F_{\text{local}} \in [B, N_{\text{stage}}, K, D_{\text{stage}}]$
    Attentive Aggregate($F_{\text{local}}$)
    Get $F_{\text{aggre}} \in [B, N_{\text{stage}}, D_{\text{stage}}]$
    Global Extractor($F_{\text{aggre}}$)
    Get $PN = F_{\text{global}} \in [B, N_{\text{stage}}, D_{\text{stage}}]$
end for
A-Bi-LSTM
Forward Get $y_t \in [B, N_3, D_{S_{\text{num}}}/2]$
Reverse Get $y'_t \in [B, N_3, D_{S_{\text{num}}}/2]$
Attention Get $Y_a \in [B, D_{S_{\text{num}}}]$
Regressor
Get 6-DOFs pose $(\hat{p}, \hat{q})$
```

4.1. Dataset

We employ the widely evaluated event-based CPR dataset IJRR [23] and M3ED [4] , encompassing both indoor and outdoor scenes. Two distinct methods to partition the CPR dataset [26] have been benchmarked: random split and novel split. In the random split approach, the dataset is randomly selected 70% of all sequences for training and allocated the remaining sequences for testing. On the other hand, in the novel split, we divide the data chronologically, using the initial 70% of sequences for training and the subsequent 30% for testing.

4.2. Baseline

We perform a thorough evaluation of our proposed method by comparing it with SOTA event-based approaches, namely CNN-LSTM [37] and AECRN [18]. Moreover, we present results derived from other well-established computer vision methods, including PoseNet[14], Bayesian PoseNet [13], Pairwise-CNN [15], LSTM-Pose [39], and SP-LSTM[26].

4.3. IJRR Dataset Results

4.3.1 Random Split Results

Based on the findings presented in Tab. 1, it is apparent that PEPNet surpasses other models concerning both rotation and translation errors across all sequences. Notably, PEPNet achieves these impressive results despite utilizing significantly fewer model parameters and FLOPs compared to the frame-based approach. Moreover, PEPNet not only exhibits a remarkable 38% improvement in the average error compared to the SOTA CNN-LSTM method but also attains superior results across nearly all sequences.In addressing the more intricate and challenging hdr_poster sequences, while the frame-based approach relies on a denoising network to yield improved results [12], PEPNet excels by achieving remarkable performance without any additional processing. This observation strongly implies that PEPNet's Point Cloud approach exhibits greater robustness compared to the frame-based method, highlighting its inherent superiority in handling complex scenarios.

Furthermore, we introduce an alternative variant, PEPNet$_{tiny}$, which integrates a lighter model architecture while preserving relatively strong performance. As depicted in Fig. 3, PEPNet consists of three stages, and the model's size is contingent upon the dimensionality of MLPs at each stage. The dimensions for the standard structure are [64, 128, 256], whereas those for the tiny structure are [16,

Network	PoseNet	Bayesian PoseNet	Pairwise-CNN	LSTM-Pose	SP-LSTM	CNN-LSTM	PEPNet	PEPNet$_{tiny}$
Parameter	12.43M	22.35M	22.34M	16.05M	135.25M	12.63M	0.774M	**0.064M**
FLOPs	1.584G	3.679G	7.359G	1.822G	15.623G	1.998G	0.459G	**0.033G**
shapes_rotation	0.109m,7.388°	0.142m,9.557°	0.095m,6.332°	0.032m,4.439°	0.025m,2.256°	0.012m,1.652°	**0.005m,1.372°**	0.006m,1.592°
box_translation	0.193m,6.977°	0.190m,6.636°	0.178m,6.153°	0.083m,6.215°	0.036m,2.195°	**0.013m**,0.873°	0.017m,**0.845°**	0.031m,1.516°
shapes_translation	0.238m,6.001°	0.264m,6.235°	0.201m,5.146°	0.056m,5.018°	0.035m,2.117°	0.020m,1.471°	**0.011m,0.582°**	0.013m, 0.769°
dynamic_6dof	0.297m,9.332°	0.296m,8.963°	0.245m,5.962°	0.097m,6.732°	0.031m,2.047°	0.016m,1.662°	**0.015m,1.045°**	0.018m,1.144°
hdr_poster	0.282m,8.513°	0.290m,8.710°	0.232m,7.234°	0.108m,6.186°	0.051m,3.354°	0.033m,2.421°	**0.016m,0.991°**	0.028m,1.863°
poster_translation	0.266m,6.516°	0.264m,5.459°	0.211m,6.439°	0.079m,5.734°	0.036m,2.074°	0.020m,1.468°	**0.012m,0.588°**	0.019m,0.953°
Average	0.231m,7.455°	0.241m,7.593°	0.194m,6.211°	0.076m,5.721°	0.036m,2.341°	0.019m,1.591°	**0.013m,0.904°**	0.019m,1.306°

Table 1. IJRR random split results. The table presents the median error for each sequence, as well as the average error across the six sequences. It also presents the number of parameters and FLOPs for each model. Bold indicates the most advanced result, while underline signifies the second-best result.

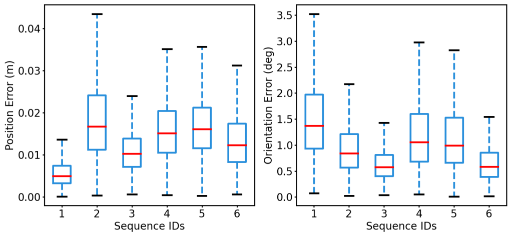

Figure 5. Error distribution of event-based CPR results achieved by PEPNet using a random split. (a) Translation errors. (b) Rotation errors.

32, 64]. As indicated in Tab. 1, even with a mere 0.5% of the CNN-LSTM's parameter, PEPNet$_{tiny}$ achieves comparable and even slightly superior results. This remarkable outcome emphasizes the superiority of leveraging event cloud data processing directly.

4.3.2 Error Distribution

Fig. 5 illustrates the error distribution of PEPNet across six distinct sequences using the random split method, specifically: shape rotation, box translation, shape translation, dynamic 6-dof, hdr poster, and poster translation. To enhance clarity, the top and bottom boundaries of the box represent the first and third quartiles, respectively, indicating the interquartile range (IQR). The median is denoted by the band within the box. It is observed that the IQR of the translation error approximately locates between 0.004m and 0.024m, while the orientation error ranges from 0.4° to 1.9°.

4.3.3 Novel Split Results

To assess the model's robustness, we adopt the novel split as an evaluation criterion, as shown in Tab. 2. During the training process, we observe a more pronounced overfitting phenomenon in PEPNet compared to the random split. We attribute this observation to the disparities in data distributions between the trainset and the testset, as well as the limited data size. Contrary to the methods we compared, PEPNet does not necessitate pre-trained weights. For instance, SP-LSTM relies on pre-trained VGG19 weights from Imagenet, while AECRN requires synthetic heuristic depth and an extensive pretraining process.

To address overfitting, PEPNet employs conventional methods that yield consistent and comparable results with the SOTA on three shape sequences that are displayed in the network column of Tab. 2. It is essential to note that AECRN adopts a hybrid approach, combining neural network regression for scene coordinates with derivable RANSAC for pose estimation. Moreover, this method incurs significant time consumption, with even the SOTA DSAC* algorithm taking nearly 30ms, excluding additional time for data format conversion. This time constraint presents compatibility challenges with the low-latency nature of event cameras. In contrast, PEPNet can execute on a server in just 6.7ms, with the main time-consuming module being grouping and sampling. Furthermore, with potential field programmable gate array (FPGA) or application-specific integrated chip (ASIC) support for these operations[6, 20], PEPNet's performance can be further accelerated.

4.4. M3ED Dataset Results

We selected three robots (Car, Falcon, and Spot) to extend the application scope of PEPNet across five sequences in an outdoor night setting, as illustrated in the Tab. 3. Due to its much higher resolution than IJRR, we performed downsampling processing and more number of points (1024 to 2048), and other experimental configurations are consistent with the IJRR dataset with random split. The results demonstrate the superior performance of PEPNet even in more challenging outdoor environments.

4.5. Attention Visualization

As shown in Fig. 6, We observe that the attention scores exhibit larger at both the beginning and end. We tentatively infer that the model focuses more on the difference in features between the start and the end for CPR, which is also

Network	PoseNet	Bayesian PoseNet	Pairwise-CNN	LSTM-Pose	SP-LSTM	DSAC*	AECRN	**PEPNet**
shapes_rotation	0.201m,12.499°	0.164m,12.188°	0.187m,10.426°	0.061m,7.625°	0.045m,5.017°	0.029m,2.3°	0.025m,2.0°	**0.016m,1.745°**
shapes_translation	0.198m,6.696°	0.213m,7.441°	0.225m,11.627°	0.108m,8.468°	0.072m,4.496°	0.038m,2.2°	0.029m,1.7°	**0.026m,1.659°**
shapes_6dof	0.320m,13.733°	0.326m,13.296°	0.314m,13.245°	0.096m,8.973°	0.078m,5.524°	0.054m,3.1°	0.052m,3.0°	**0.045m,2.984°**
Average	0.240m,11.067°	0.234m,10.975°	0.242m,11.766°	0.088m,8.355°	0.065m,5.012°	0.040m,2.53°	0.035m,2.23°	**0.029m,2.13°**
Inference time	5ms	6ms	12ms	9.49ms	4.79ms	30ms	30ms	**6.7ms**

Table 2. IJRR novel split results. Referred to as Tab. 1, showcases identical information. To assess the model's runtime, we conduct tests on a server platform, specifically focusing on the average time required for inference on a single sample.

M3ED	PoseNet	LSTM-Pose	CNN-LSTM	**PEPNet**
INPUT	Event Frame	Event Frame	Event frame	**Point Cloud**
Falcon_Night_High_Beams	0.181m,2.221°	0.112m,0.946°	0.107m,1.435°	**0.082m,0.575°**
Car_Night_Pen_S_Loop	1.618m,8.126°	0.667m,4.914°	0.773m,3.005°	**0.577m,1.319°**
Spot_Night_Pen_Loop	1.735m,5.502°	0.761m,7.898°	**0.401m**,1.771°	0.468m,**1.062°**
Car_Pen_S_Loop_darker	1.841m,4.575°	0.751m,3.738°	0.598m,2.772°	**0.385m,1.01°**
Spot_Plaza_Light	1.372m,9.564°	0.565m,5.221°	**0.273**m,2.001°	0.348m,**1.234°**
Avergae	1.349m,5.998°	0.571m,4.543°	0.43m,2.197°	**0.372m,1.04°**

Table 3. Outdoor extension on M3ED dataset with random split.

Condition	HS	LSTM	Bi-LSTM	Aggregation	Translation	Rotation	T+R
1	✓			Max	0.015m	0.884°	3.04
2	✓			Temporal	0.014m	0.786°	2.77
3	✓	✓		Max	0.014m	0.833°	2.85
4	✓	✓		Temporal	0.012m	0.603°	2.25
5	✓		✓	Max	0.014m	0.813°	2.82
6	✓		✓	Temporal	**0.011m**	**0.582°**	**2.12**

Table 4. Abalation Study for three key modules. T+R = Translation + Rotation·$\pi/180$ (m+rad)

Scence	$\alpha = 0.5, \beta = 0.5$	$\alpha = 0.25, \beta = 0.75$	$\alpha = 0.75, \beta = 0.25$
shape_translation	**0.0302m,1.684°,5.96**	0.0359m,1.72°,6.59	0.0303m,2.056°,6.62
shape_rotation	0.0143m,2.888°,6.47	**0.0159m,2.68°,6.27**	0.014m,3.36°,7.26
dynamic_6dof	**0.0542m,2.799°,10.3**	0.0611m,2.488°,10.5	0.0516m,3.251°,10.8

Table 5. Abalation Study for loss function's coefficient.

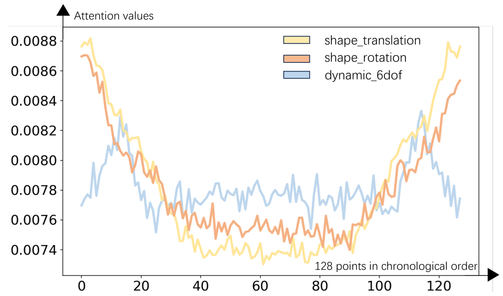

Figure 6. Visualization of the attention values in the time domain. 128 points in chronological order on the horizontal axis and the attention values of the corresponding point on the vertical axis.

seen in the geometry approach [7, 24].

4.6. Ablation Study

Key Module Ablation: In order to validate the efficacy of key modules, we conducted an ablation experiment focusing on three primary components: hierarchy structure, Bi-LSTM, and temporal aggregation. These experiments are designed to evaluate rotation and translation errors on the shape translation sequence with the random split. The combined error (T+R) is measured after processing. Our experimental setup comprises four distinct conditions, as illustrated in Tab. 4. Condition 1 represents the sole utilization of the hierarchy structure (HS), while Condition 2 combines the ordinary LSTM. Condition 3 incorporates the bidirectional LSTM, and Condition 4 integrates the attention mechanism for feature aggregation. The ablation experiments reveal significant insights. Experiments 1 and 3 demonstrate that augmenting LSTM enhances the extraction of explicit temporal features. Moreover, experiments 3 and 5 reveal the effectiveness of the bidirectional LSTM in extracting motion information. Additionally, experiments 5 and 6 confirm the notable impact of attention in feature aggregation, resulting in a substantial reduction in error rates.

Loss ablation: We incorporated the experiment involving scaling coefficients of the loss function in Tab. 5. This experiment utilized a tiny version of PEPNet, trained for 100 epochs, and the outcome is MSE in translation, rotation, and T+R. Across three distinct motion scenarios (translation, rotation, and 6dof) varied coefficient ratios induced deviations in the obtained results. For example, in shape rotation, increasing the weight on rotation makes the results better.

5. Conclusion

In this paper, we introduce an end-to-end CPR network that operates directly on raw event clouds without frame-based preprocessing. PEPNet boasts an impressively lightweight framework that adeptly extracts spatial and temporal features, leading to SOTA performance. Diverging from frame-based approaches, our method prioritizes preserving the inherent distribution of the event cloud, capitalizing on its sparse nature to achieve extraordinary capabilities for ultra-low-power applications.

Acknowledgment. This work was supported in part by the Young Scientists Fund of the National Natural Science Foundation of China (Grant 62305278), as well as the Hong Kong University of Science and Technology (Guangzhou) Joint Funding Program under Grant 2023A03J0154 and 2024A03J0618.

References

[1] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–767, 2018. 2

[2] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5847–5865, 2021. 2

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6684–6692, 2017. 2

[4] Kenneth Chaney, Fernando Cladera, Ziyun Wang, Anthony Bisulco, M Ani Hsieh, Christopher Korpela, Vijay Kumar, Camillo J Taylor, and Kostas Daniilidis. M3ed: Multi-robot, multi-sensor, multi-environment event dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4015–4022, 2023. 2, 6

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3

[6] Haotian Fu, Yulong Huang, Tingran Chen, Chenyi Fu, Hongwei Ren, Yue Zhou, Shouzhong Peng, Zhirui Zong, Biao Pan, and Bojun Cheng. Ds-cim: A 40nm asynchronous dual-spike driven, mram compute-in-memory macro for spiking neural network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024. 7

[7] Guillermo Gallego, Christian Forster, Elias Mueggler, and Davide Scaramuzza. Event-based camera pose tracking using a generative event model. *arXiv preprint arXiv:1510.01972*, 2015. 8

[8] Guillermo Gallego, Jon EA Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2402–2412, 2017. 2

[9] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 2

[10] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[12] Yifan Jin, Lei Yu, Guangqiang Li, and Shumin Fei. A 6-dofs event-based camera relocalization system by cnn-lstm and image denoising. *Expert Systems with Applications*, 170: 114535, 2021. 2, 6

[13] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016. 6

[14] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2, 6

[15] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 929–938, 2017. 2, 6

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2

[17] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43 (2):566–576, 2008. 1

[18] Hu Lin, Meng Li, Qianchen Xia, Yifeng Fei, Baocai Yin, and Xin Yang. 6-dof pose relocalization for event cameras with entropy frame and attention networks. In *The 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 1–8, 2022. 2, 6

[19] Yimin Lin, Zhaoxiang Liu, Jianfeng Huang, Chaopeng Wang, Guoguang Du, Jinqiang Bai, and Shiguo Lian. Deep global-relative networks for end-to-end 6-dof visual localization and odometry. In *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part II*, pages 454–467. Springer, 2019. 2

[20] Haobo Liu, Zhengyang Qian, Wei Wu, Hongwei Ren, Zhiwei Liu, and Leibin Ni. Afpr-cim: An analog-domain floating-point rram-based compute-in-memory architecture with dynamic range adaptive fp-adc. *arXiv preprint arXiv:2402.13798*, 2024. 7

[21] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *International Conference on Learning Representations*, 2021. 3

[22] Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermuller, and Yiannis Aloimonos. Learning visual motion segmentation using event surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14414–14423, 2020. 2

[23] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1, 2, 6

[24] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry

for event cameras. *IEEE Transactions on Robotics*, 34(6): 1425–1440, 2018. 8

[25] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530. IEEE, 2017. 2

[26] Anh Nguyen, Thanh-Toan Do, Darwin G Caldwell, and Nikos G Tsagarakis. Real-time 6dof pose relocalization for event cameras with stacked spatial lstm networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 6

[27] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2010. 1

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 3

[30] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018. 2

[31] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018. 2

[32] Hongwei Ren, Yue Zhou, Haotian Fu, Yulong Huang, Renjing Xu, and Bojun Cheng. Ttpoint: A tensorized point cloud network for lightweight action recognition with event cameras. *arXiv preprint arXiv:2308.09993*, 2023. 2

[33] Hongwei Ren, Yue Zhou, Yulong Huang, Haotian Fu, Xiaopeng Lin, Jie Song, and Bojun Cheng. Spikepoint: An efficient point-based spiking neural network for event cameras action recognition. *arXiv preprint arXiv:2310.07189*, 2023.

[34] Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. Eventnet: Asynchronous recursive event processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3887–3896, 2019. 2

[35] Yoli Shavit and Ron Ferens. Introduction to camera pose estimation with deep learning. *arXiv preprint arXiv:1907.05272*, 2019. 1

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[37] Ahmed Tabia, Fabien Bonardi, and Samia Bouchafa. Deep learning for pose estimation from event camera. In *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7. IEEE, 2022. 6

[38] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6939–6946. IEEE, 2018. 2

[39] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 627–637, 2017. 2, 6

[40] Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Space-time event clouds for gesture recognition: From rgb cameras to event cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1826–1835. IEEE, 2019. 2, 3

[41] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving deeper into convolutional neural networks for camera relocalization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651. IEEE, 2017. 2

[42] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 3

[43] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3323–3332, 2019. 3

[44] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 3