Encryption of Event Camera Data for Visual Localisation

How can the encryption of raw event camera data be practically and effectively used for privacy protection in a visual localisation application?

Bink Boëtius

Supervisor(s): Nergis Tömen, Tunahan Parlayici

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2026

Name of the student: Bink Boëtius
Final project course: CSE3000 Research Project
Thesis committee: Nergis Tömen, Tunahan Parlayici, Ricardo Marroquim

Abstract

Event cameras are bio-inspired sensors that asynchronously measure per-pixel brightness changes, offering lower power consumption and higher temporal resolution than conventional frame cameras. These properties make them suitable for privacy-sensitive applications, such as visual localisation in AR/VR systems, where client-server architectures are used to offload computationally expensive processing from resource-limited edge devices. However, transmitting visual data to a service provider introduces privacy risks. Kim et al. propose a privacy-preserving visual localisation method that assumes an honest-but-curious service provider, but acknowledge that their approach is insufficient against a more capable attacker that can, for example, extract raw event data directly. This paper addresses this limitation by encrypting raw event camera data using the algorithm described by Zhang et al., for which no implementation was previously available. The algorithm is implemented within the visual localisation pipeline of Kim et al. and evaluated on the EvRooms dataset. The theoretical and practical effectiveness of the encryption is analysed, and improvements to the original algorithm are proposed and tested. The impact on both privacy preservation and localisation performance is measured. The paper shows that the polarity-mapping step in the implemented encryption algorithm is a powerful event data obfuscation process while still allowing retrieval of the original data. However, this process is currently not dependent on a key, which makes the algorithm not secure according to Kerckhoffs's principle. Further research should explore encryption algorithms that employ key-dependent polarity mapping. The code used in this research can be found on GitHub.

1 Introduction

Event-based vision is an emerging field that focuses on implementing computer vision using event cameras. These are bio-inspired sensors that asynchronously measure per-pixel brightness changes. These sensors only send a signal when a change in brightness is significant, which results in lower power consumption than conventional frame cameras that signal a value for each pixel on every frame. At the same time, the sensors of an event camera are not bound by an internal clock, resulting in significantly higher temporal resolution [1]. Using this technology, some difficult tasks for conventional cameras can be tackled, such as filming in low-light conditions and filming fast motion [2] [3]. Since event cameras do not process full RGB frames, they are a promising technology for privacy-sensitive applications. However, while not all visual data is processed and stored, some information can still be inferred from the collected data [4].

In VR and AR applications, event-based vision can be used to localise the user in a pre-built environment. Capturing the visual data is then possible without consuming much energy or requiring much memory. However, localising the user using this data still requires substantial computational power, which most AR/VR devices, such as smartphones or AR glasses, cannot provide. Because of this, client-server localisation is generally performed, which possibly reveals private information to the server. On top of that, some people might not want to be captured on camera entirely. To minimise potential privacy concerns, Kim et al. [5] propose a privacy-preserving visual localisation method. Here, it is assumed that the service provider is honest-but-curious, meaning that they faithfully handle the data as expected, but may extract visual data from the client if possible. The authors note that, against a more potent attacker who can, for example, extract the raw event data, their privacy-preserving methods are inadequate. They proposed that adding encryption to the event data would help minimise these security risks.

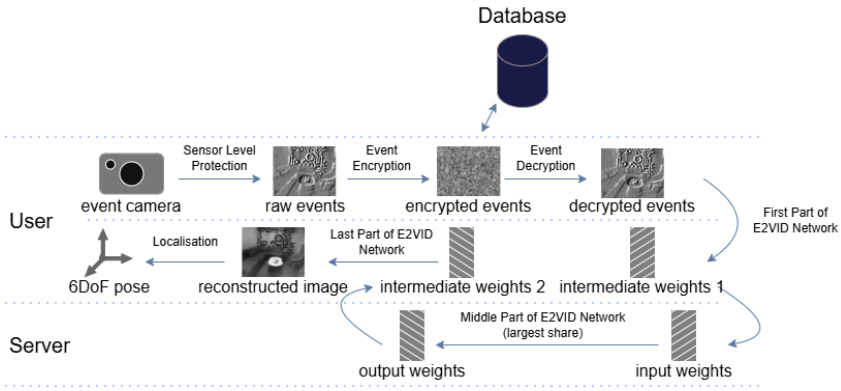


Figure 1: The entire visual localisation pipeline analysed in this paper. This pipeline is mostly copied from Kim et al. [5], including the sensor-level protection and the splitting of the event-to-image network, which, in this case, is E2VID. The addition is the event encryption and decryption based on the algorithm proposed by Zhang et al. [6].

Since an honest-but-curious service provider cannot always be assumed, and other attackers may attempt to exploit AR and VR systems, this paper aims to enhance the security of these systems. This is done by encrypting the event data captured on the AR and VR devices using the algorithm described by Zhang et al. [6]. No implementation of this algorithm was provided, so I implemented it in the codebase created by Kim et al. for visual localisation, as shown in Figure 1.

The implementation is tested and analysed to answer the following research question:

"How can the encryption of raw event camera data be practically and effectively used for privacy protection in a visual localisation application?"

To this end, it is first explored how the previously mentioned algorithm can successfully be implemented in the visual localisation pipeline. The theoretical and practical effectiveness of the algorithm will be analysed and discussed to determine its usefulness. After that, the impact of this method on both

the preserved privacy and the task performance will be analysed. Possible adjustments to improve the algorithm will also be tested.

This paper's contributions are as follows:

- It shows that the polarity-mapping step in the encryption algorithm designed by Zhang et al. is the dominant contributor to data obfuscation.
- It states that the polarity-mapping step currently does not depend on any key and is completely reversible by any attacker. Therefore, it does not comply with Kerckhoffs's principle for secure encryption.
- It suggests that a data encryption algorithm that uses only key-dependent polarity-mapping could improve on the current algorithm by Zhang et al. and should be explored in further research.
- It reevaluates the effects of both camera motion and brightness on localisation performance. It extends this to the effects on privacy performance after encryption. However, it acknowledges that the data contains too many variables to draw strong conclusions.

Related work will be discussed in Chapter 2. Chapter 3 will elaborate upon the implementation of the raw event encryption in the visual localisation pipeline. Chapter 4 covers the improvements that were made to the algorithm of Zhang et al. [6] and why these were made. In Chapter 5, the methods used to evaluate the privacy and the task performance will be explained, and the results of these experiments will be visualised. The results will be discussed in a broader context in Chapter 6. In Chapter 7, conclusions are drawn from the research, and possible shortcomings are noted. Further research is also proposed in this chapter. Chapter 8 reflects on the ethical aspects of the research and the reproducibility of the methods used.

2 Related Work

2.1 Event cameras

Event cameras were created to mimic the behaviour of the retina and the way the brain processes these inputs, and have been around for some decades [7]. Since then, multiple variants of event cameras have been introduced, including [8], [9], [10], and [11]. Research has been conducted to evaluate this new technology across multiple use cases. Event cameras have, for example, been used to enhance video super-resolution [12] [13], and enable lossless video compression [9]. Furthermore, extensive research has been conducted on detection and recognition using event cameras. For example, research has been conducted on gait, a person's manner of walking, and gesture recognition [14] [15] [16], on recognising potential hazards on the road [17], and other object detection and recognition algorithms [18] [19] [20] [21].

2.2 Event-Based Visual Localisation

The application this paper focuses on is visual localisation. A lot of research has already been done on using event cameras to reconstruct their environment and estimate a 6-degree-of-freedom (translation and rotation) pose of a camera [22] [23] [24] [25]. Visual localisation differs from this in that it estimates a pose within a pre-built map. Research on visual localisation has been done using intensity cameras for both building the map and localising the camera [26], as well as using intensity cameras only for building the map and doing localisation with event cameras [27] [28].

In this paper, the visual localisation implementation of Kim et al. [5] is used, which employs event cameras for both map construction and localisation. According to Kim et al., using raw events directly for localisation [29] [30] [31] usually performs worse than using structure-based methods, which compare feature descriptors to get a pose. It is noteworthy that the translation and rotation errors measured by Kim et al. for these direct methods were significantly worse than those reported in the original papers. The difference in datasets used might be the reason for this difference. The visual localisation performed by Kim et al. uses an event-to-image conversion neural network to represent the event data. They concluded that this yields better results than using other raw event representations [32] [16] or more light-weight image reconstruction methods [33] [34]. When using the other structure-based methods, it should be noted that the pre-built 3D maps were still made using event-to-image conversion, which could have been favourable for the localisation process using the same method. Future research could investigate whether different representations of the 3D maps would yield different results for different event representations.

2.3 Event encryption

Since conventional intensity images and videos can contain a lot of private information, a lot of research has been conducted on the encryption of these media using chaotic systems [35] [36] [37] [38] or statistical methods [39] [40] [41]. In prior work, it was discussed that raw data from event cameras contained less private information [4]. However, the way this is explained does not hold up under careful examination, since the privacy preservation is measured by how well a person can be re-identified through reconstructed images. This is insufficient as a privacy metric, since the authors themselves show that their proposed method can re-identify individuals directly from raw event data. If the raw data enables re-identification, it contains personally identifiable information by definition, regardless of how a separate reconstruction attack performs. What the reconstruction attack results actually show is that re-identification from reconstructed images is less accurate than from standard intensity images, and than using the method described by the authors. An attacker could use their method for re-identification rather than image reconstruction, thereby defeating the claim of privacy preservation. More research further shows that privacy-sensitive information can be inferred from raw event data. From raw events, faces and facial expressions can be detected [42] [43], and individuals can be re-identified using multiple techniques [4] [14] [15]. Moreover, methods have been developed to reconstruct raw events into high-quality images and videos [44] [45] [46] [47].

This raises the demand for privacy-protection methods for event data. However, not much research has been conducted on this topic. One example would be the E2PRIV model [48],

which reconstructs videos from events while obscuring facial details. The visual localisation process of Kim et al. also obfuscates facial details at the sensor level [5], since they are not needed for visual localisation. While this prevents attackers from retrieving facial information even if they obtain raw events, it still preserves other privacy-sensitive information captured in the environment. This is relevant, since AR and VR applications that would depend on visual localisation would also be used in private spaces. Another approach would be to remove any privacy-sensitive data from the event stream while retaining the data needed for the downstream task [49]. However, this would not be practically feasible if the data needed for the downstream task is also privacy-related. While this has not been tested, this might be the case for visual localisation when trying to hide the layout of a private space. To ensure that all data is being retained and does not leak any private information, it should be encrypted. To my knowledge, only Du et al. [50] and Zhang et al. [6] have explored direct encryption and decryption of raw event data. In this paper, the implementation by Zhang et al. will be examined in greater detail.

2.4 Privacy measurements

To measure the privacy preservation of the encryption algorithm implemented in this paper, images reconstructed from the encrypted events are compared with those reconstructed from the original events. The more they differ, the less privacy-sensitive information is leaked. This difference can be objectively measured using different approaches, such as PSNR, MSE, SSIM, and FSIM [51]. However, the results of these tests do not always align with human perception, which is important when discussing the privacy of visual media. To measure privacy in a way that more closely resembles human perception, learning-based methods have been developed. LPIPS [52] is trained to recognise whether images are similar overall, while SemSim [53] is specifically trained to recognise if key features of an image are still identifiable.

3 Methodology

3.1 Initial localisation setup

Since this paper elaborates on the visual localisation pipeline by Kim et al. [5], the remaining encryption and decryption logic has been implemented within their provided codebase. The implementation of the localisation pipeline depended on some hardcoded file paths. I changed this to depend more on input flags, so that sources and destinations could also reside in different places. A small piece of code was added to reformat the localisation results into CSV format, so they could be inspected more easily. Another piece of code was added to enable measures across multiple scenes in the dataset, rather than just one. Finally, the code has been slightly altered to save reconstructed images during the localisation process, so they can later be used to take privacy measurements for encrypted events.

To measure the performance of visual localisation, Kim et al. use the DAVIS240C dataset [54] and their own EvRooms dataset. These were grouped into scenes with 'normal', 'fast-moving', and 'low-light' conditions. After further inspection, these splits did not seem accurate, so I analysed the event streams to make new splits. I omitted the DAVIS240C dataset since its format was incompatible with the visual localisation implementation, and Kim et al. did not explain how they converted it. To examine the lighting conditions of the EvRooms dataset, I measured the average frame intensity of frames captured by the DAVIS346 camera [55], which captures both event and intensity frames. For the perceived motion, I measured the average density of events in batches of 30000, the same number used to create an event voxel from raw events in the localisation pipeline. The results from these analyses, as shown in Table 1, showed that the lighting conditions and perceived motion of the event streams formed a more continuous than a discrete distribution. That is why localisation performance and privacy preservation are measured per scene and shown relative to the lighting conditions and perceived motion in Chapter 5. From these measurements, it became apparent that 'room_2' is identical to 'room_1', so it is omitted.

Table 1: Average brightness and motion per scene. The brightness is measured as the average pixel intensity across all captured frames in a scene. The motion is measured as the average event density (events / second) across batches of 30000 events in the event stream.

Scene	**Avg. Brightness** (Pixel intensity (0-255))	**Avg. Motion** (Events (k) / Second)
boxes	55.6	1,600
cabinet	48.7	3,442
cabinet_dark	28.7	1,810
desk	108.8	3,936
desk_dark	7.9	3,520
fan	65.5	2,938
kitchen_1	38.7	1,550
kitchen_2	37.3	1,212
lounge	100.8	2,602
office_301	105.3	3,254
office_inmc	72.1	2,218
office_inmc_dark	47.9	1,504
paris	45.5	1,390
pingpong_1	44.2	1,294
playroom	58.3	1,341
pullup	52.1	2,053
pullup_dark	40.1	1,081
robot	65.8	2,576
robot_dark	133.2	2,897
room_1	39.5	2,041
room_3	35.3	1,594
sofa	51.9	3,016
sofa_dark	38.2	1,417
table	53.3	4,340
table_dark	0.9	2,290

3.2 Implementing raw event encryption

Since no implementation of the encryption and decryption algorithm by Zhang et al. [6], shown in Figure 2, has been shared, I implemented it in the visual localisation codebase. This implementation is visualised in Figure 3. Scenes captured by events are first split into multiple chunks, and each chunk is encrypted individually. For each chunk, a spatial projection, $\mathbf{E_x}$, of all events is created, together with a mask, $\mathbf{M}$, that contains all coordinates not present in the spatial pro-

Algorithm 1: Event Encryption

Input : $\mathbf{E}$
Output: $\tilde{\mathbf{E}}$

Initialize $\mathbf{E_x} = \{\mathbf{x}_i\}_{i=1:I}$ from $\mathbf{E}$, and $\tilde{\mathbf{E}} = \mathbf{E}$;
Initialize `mask` $\mathbf{M}$ via `function` δ
foreach $\mathbf{x}_i \in \mathbf{E_x}$ **do**
  Find `spatial neighbors` $\mathbf{R} = \theta(\mathbf{x}_i, \mathbf{M})$
  if $\mathbf{R} = \varnothing$ **then continue**
  Find `central events` $\mathbf{E}_c = \{\mathbf{e} \in \mathbf{E} \mid \mathbf{x} = \mathbf{x}_i\}$
  Synthesize `noise` $\mathbf{N} = \phi(\mathbf{R}, \mathbf{E}_c)$
  $\tilde{\mathbf{E}} = \tilde{\mathbf{E}} \cup \mathbf{N}$
  $\mathbf{E_x} = \mathbf{E_x} \cup \mathbf{R}$
  $\mathbf{M} = \mathbf{M} \setminus \mathbf{R}$
  if $\mathbf{M} = \varnothing$ **then break**
Compute `polarity mapping` $\tilde{\mathbf{E}} = \lambda(\tilde{\mathbf{E}})$

Figure 2: Encryption algorithm designed by Zhang et al. [6].

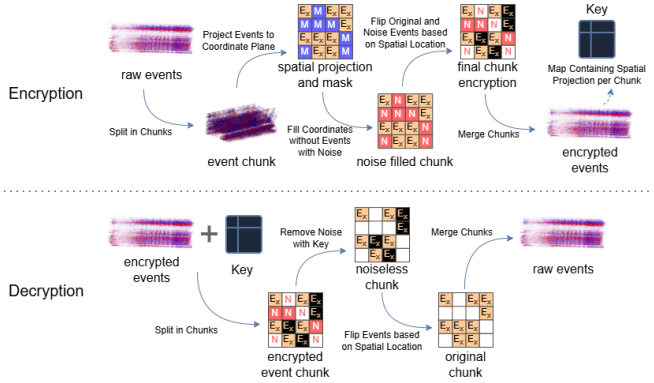


Figure 3: Visualisation of my implementation of the encryption algorithm designed by Zhang et al. [6]. At the top, the encryption process is shown, where the raw event stream is split into chunks. A spatial projection $\mathbf{E_x}$ is created from the events with the mask $\mathbf{M}$ consisting of the leftover coordinates. The mask is filled with noise, and afterwards, all events are either flipped or not based on their spatial location. The chunks are merged to create an encrypted event stream. The key is a map that contains the spatial projection $\mathbf{E_x}$ for each chunk, indicated by the timestamp of the first event in the next chunk. On the bottom, the decryption process is visualised. Here, the encrypted event stream is split into chunks based on the key's timestamps. Afterwards, the key is used to remove all noise from the chunks. The polarities are flipped again based on the spatial location of the events, and the chunks are finally merged to recreate the original event stream.

jection. According to Zhang et al., any mask that does not overlap with the spatial projection works; however, this mask configuration was chosen because it was also used by Zhang et al., was the most straightforward to implement, and allows for the most noise to be injected. Whether this is an optimal mask could be researched in future work. The spatial projection will be stored as a key to decrypt the events later.

According to Zhang et al., noise is synthesised as follows. For each spatial location in the spatial projection, the spatial neighbours, $\mathbf{R}$, that fall within the mask are computed.

$$\theta(\mathbf{x}_i, \mathbf{M}) = \{\mathbf{x} \in \mathbf{M} \mid f_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_i) < T_{\mathbf{x}}\}, \tag{1}$$

In Equation (1), $f_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_i)$ computes the Manhattan distance [56] between an arbitrary $\mathbf{x}$ in the mask and the spatial location $\mathbf{x}_i$. Just like the authors, I implemented $T_{\mathbf{x}}$ to be 2, so that spatial neighbours can only be the directly horizontal and vertical neighbouring pixel of each $\mathbf{x}_i$ in $\mathbf{E_x}$. For each event in $\mathbf{x}_i$, called central events $\mathbf{E}_c$, a noise event in all of its spatial neighbours is created. According to Zhang et al., these noise events, $\mathbf{N}$, should have the following properties

$$\begin{cases} \mathbf{x} \in \mathbf{R} \\ p \in \{-1, +1\} \\ t = t_i \left(1 + \sigma f_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_i)\right) \\ f_t(t, t_i) < T_t \end{cases}, \tag{2}$$

where $\mathbf{x}$ is the spatial coordinate, p is the polarity, and t is the timestamp of the noise event. Moreover, $\mathbf{x}_i$ and $\mathbf{t}_i$ denote the coordinate and timestamp of the central event. The σ is a scaling factor, $f_t(t, t_i)$ computes the time interval between the timestamp of the noise event and the central event, and T_t is an upper threshold for this interval. The last two properties might conflict, especially as timestamps increase, resulting in larger intervals. Zhang et al. set σ to 0.05 and made T_t variable. This effectively means the fourth property is not used, so it was omitted in the implementation. Note that timestamps that would exceed the first timestamp of the next chunk are omitted, since noise from one chunk would otherwise bleed into the next chunk. This might introduce noise into the spatial projection of the next chunk, which cannot be removed during decryption.

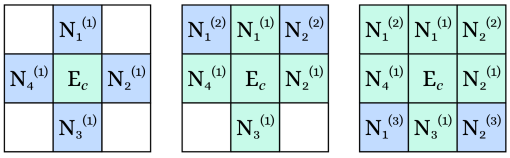


Figure 4: The recursive noise synthesisation described by Zhang et al. [6], where only direct horizontal and vertical adjacent pixels are recognised as spatial neighbours. On the left, the first iteration is shown, where noise events in the spatial neighbours $\mathbf{N}_i^{(1)}$ $(i = 1, 2, 3, 4)$ are created from the central events $\mathbf{E}_c$ where $|\ \mathbf{N}_i^{(1)}\ | = |\ \mathbf{E}_c\ |$. The middle grid shows a possible second iteration when the $\mathbf{N}_1^{(1)}$ noise events are chosen as the next central events. Here $|\ \mathbf{N}_i^{(2)}\ | = |\ \mathbf{N}_1^{(1)}\ |$. On the right, the result after a possible third, and in this case last, iteration is shown, where the $\mathbf{N}_3^{(1)}$ noise events are the new central events, and $|\ \mathbf{N}_i^{(3)}\ | = |\ \mathbf{N}_3^{(1)}\ |$

After the noise is synthesised, it is added to the encrypted events, and the spatial neighbours are added to the spatial projection and removed from the mask. This process is then repeated until the mask is empty, as shown in Figure 4. This means that all spatial locations are either filled with real events or noise events at the end of encryption.

At last, following Zhang et al., a polarity-mapping λ is performed. The polarities of both the original and noise events would be either flipped or not, depending on whether the Szudzik pairing function [57] would yield an odd or even number, respectively, when the x and y coordinates of the event were given as input.

$$\lambda(\tilde{\mathbf{E}}) = \{\mathbf{e} \in \tilde{\mathbf{E}} \mid p = p \oplus (\texttt{szudzik}(\mathbf{x}) \bmod 2)\}, \tag{3}$$

Finally, all chunks and keys are concatenated, yielding an encrypted event file and its corresponding key. The key would

be a map with chunk timestamps as keys and spatial projections as values, since the chunks would need to be split before they can be decrypted. The algorithm designed by Zhang et al. also applies the Szudzik pairing function to the spatial coordinates in the spatial projections to decrease the key size. This is omitted in this paper for practicality and because it does not make a significant difference for answering this paper's research question. Since noise events can only be synthesised after real events and before the first real event of the next chunk, the chunks can be split cleanly before decryption. To decrypt a chunk, all events with a coordinate not present in the key for that specific chunk would be removed, since these contain only noise. Afterwards, the polarity-mapping from Equation (3) is applied again, since applying the $\oplus$-operator twice results in the original value.

To measure the localisation performance, the visual localisation pipeline by Kim et al. was run on the original, encrypted, and decrypted event streams. In this pipeline, events are converted to images. These images were saved in a separate folder to test the encryption's privacy-preserving nature by comparing reconstructions from those different event streams. These measurements and results will be explained in more detail in Chapter 5.

4 Changes to the raw event encryption

The paper by Zhang et al. did not specify how many events would be encrypted simultaneously. However, this is a critical detail, as it affects the number of noise events created. This is because when a large number of events are encrypted simultaneously, their spatial projection will cover most of the pixel coordinates, leaving little room for noise events to be synthesised. For this reason, the event stream from EvRooms is split into chunks of 30000 events, the same number used to create an event voxel in the localisation pipeline, as mentioned before. Moreover, to make the algorithm slightly more efficient, a spatial event index is built before looping over the spatial projection's coordinates. Using this index, all central events for a certain coordinate can be found in $O(1)$ time instead of looping over all events for each coordinate.

Furthermore, two things stood out to me as possible flaws in the event encryption algorithm. The first one concerns the creation of timestamps of noise events. To make the timestamps relate to the original event, they are calculated using $t = t_i \left(1 + \sigma f_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_i)\right)$, as mentioned in Equation (2). This means that the distances between the noise timestamps and the original timestamps increase for events at later timestamps. This not only causes noise events to relate differently to original timestamps at different times, but also, for later events, causes more noise events to be omitted because they bleed into the next chunk. This would lead to fewer noise events overall and possibly less secure encryption. Besides, no randomness is introduced in the noise timestamps, which could reveal patterns to an attacker.

The second flaw is that the polarity mapping described by Equation (3) is reversible by any attacker aware that this algorithm is being used. This is not in accordance with Kerckhoffs's principle, which states that *"a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge"* [58], and which is widely embraced by cryptographers. The polarity mapping simply obfuscates the original data, but for an attacker knowing the algorithm, this step could just as well be omitted.

Because of this, I made three small variations to the algorithm. One where the timestamps of noise events are randomly chosen to be between 0.0 and 0.1 seconds after the original events

$$t = t_i + U\left(0.0, 0.1\right), \tag{4}$$

one where the polarity mapping, Equation (3), step is skipped, and one where both changes are made. These three variations, together with the original implementation, are tested on localisation performance and privacy preservation. The results from these measurements will be described in Chapter 5.

5 Experimental Setup and Results

5.1 Experimental setup

To measure localisation performance, the code implemented by Kim et al. [5] was used. For the 'low-light' scenes, Kim et al. used the data from the 'normal' variants of the same scenes to create the pre-built 3D maps for localisation. I modified the code so that each scene uses its own data to build the 3D map and perform localisation afterwards. This equalises the measurement conditions for all scenes. To measure privacy preservation, the implementation by Sun et al. [53] was used. For the SemSim measurements, neither the author's pre-trained weights nor the training data were available. Therefore, the semantic similarity was calculated using a ResNet-18 model pre-trained on ImageNet [59], which serves as an efficient, standardised baseline for extracting general visual features. While adequate as a replacement, the implemented version of the SemSim model differs from what the original authors envisioned.

The measurements were executed using the DelftBlue supercomputer [60]. For practical reasons, only one Intel Xeon CPU core, which uses 4 GB of system RAM, and one slice of an NVIDIA A100 GPU, which has 10 GB VRAM, were used at a time.

5.2 Localisation performance

As mentioned earlier, four different encryptions have been evaluated. For each encryption, the decrypted events are the same because the difference between the versions either affects the noise events that are deleted during decryption or the polarity mapping, which is deterministic. According to the algorithm, they should also be equal to the original event streams. But because events with the same timestamps as the first event of the next batch are decrypted in that batch, and some simultaneous events might be swapped while sorting, the decrypted event streams differ slightly from the original. Because the decrypted event streams are identical, only one result for the localisation performance of the decrypted events per scene is considered.

The localisation performance results are shown in relation to scene brightness in Figure 5 and motion in Figure 6. The localisation performance using the original event streams is

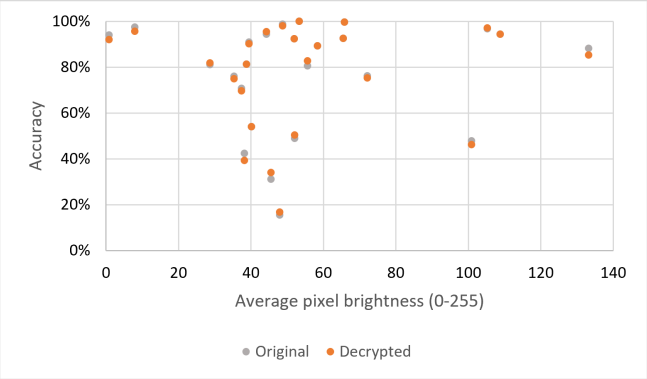


Figure 5: Localisation accuracy measured per scene. Shown in relation to the average pixel brightness of the intensity images taken during the capture of the event stream of the scene.

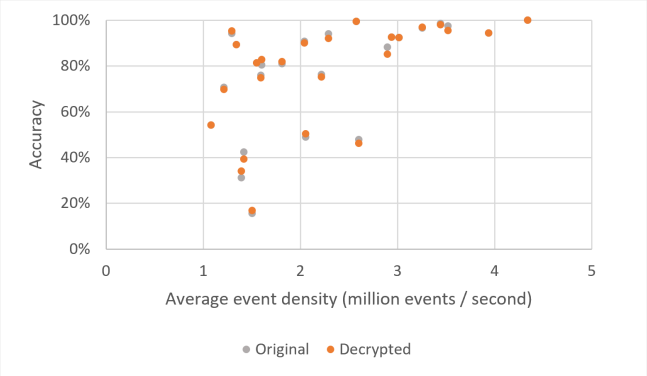


Figure 6: Localisation accuracy measured per scene. Shown in relation to the average event density, in million events per second, of the event stream. The event density is first calculated within batches of 30000 events and averaged at the end.

also displayed. It can be seen that there is no significant difference between using original and decrypted event streams, as the p-value for the paired two-tailed t-test comparing the two for each scene, performed in Excel [61], is 0.42. Furthermore, the data show a correlation between average event density and localisation performance. Figure 6 shows a scatterplot that looks like it contains a logarithmic increase. When performing a regression analysis in Excel [62] with the event density on the X range and the accuracy on the Y range, the resulting p-value is 0.004. There does not seem to be any correlation between scene brightness and localisation accuracy. There is no clear trend visible in Figure 5, and the p-value for the regression analysis is 0.737.

5.3 Privacy preservation

Comparing encryptions

For each encryption, the privacy preservation of the encrypted events is measured using multiple metrics. PSNR and MSE are used to measure the per-pixel difference between the encrypted and original reconstructed frames [51]. SSIM measures structural differences, and LPIPS and SemSim use neural networks to measure privacy preservation that more closely resemble human perception [52] [53]. The reconstructed frames from the original frames are also compared against themselves and against frames filled with only random noise, serving as baselines for privacy preservation.

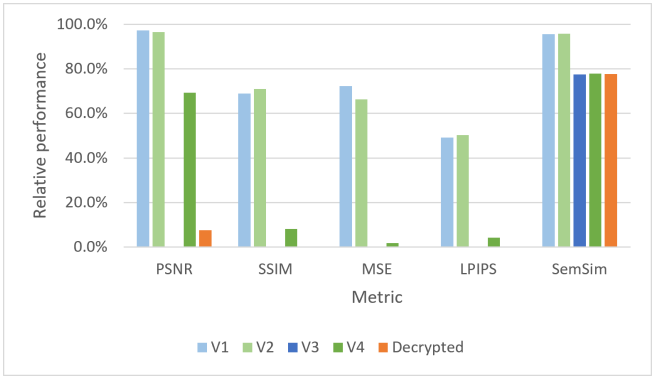
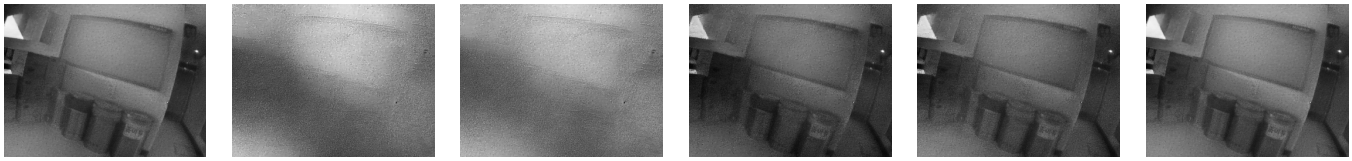


Figure 7: Average relative performance of all scenes in different privacy metrics. The base of the chart is set at the score achieved by measuring the event reconstructions of the original event stream. The top of the chart is set at the score achieved by measuring random noise. Per metric, each encryption variant is evaluated by measuring the quality of images reconstructed from the encrypted event stream. V1 uses the original algorithm. V2 uses Equation (4) for the timestamp, instead of the definition in Equation (2). V3 and V4 use the same algorithms as V1 and V2, respectively, but skip the polarity-mapping step. The decrypted events are the same across all encryption methods, so only one value is shown.

In Figure 7, the relative performance of each version of the encryption algorithm on the different privacy metrics is shown. In the Figure, the 0% line represents the result obtained when comparing images reconstructed from the original event stream with themselves. The 100% line is set to the result obtained by comparing random noise with the reconstructed images from the original event stream. For MSE, LPIPS, and SemSim, higher metric scores correspond to higher performance scores. However, for PSNR and SSIM, higher metric scores correspond to lower performance scores. It must be noted that the PSNR metric score is capped at 70 because comparing an image to itself yields infinity, and the largest value below infinity is 60.9. Since the decrypted event streams are identical across all versions, only a single value is shown for the performance of the decrypted events. It should be noted that, for the SemSim measurements, the decrypted event streams yielded different results. This is probably because of non-determinism in the hardware used by the deep neural network [63]. Since the results were similar, the average was taken to show in Figure 7.

Paired two-tailed t-tests have been performed in Excel comparing all event stream states (original, noise, V1, V2, V3, V4, decrypted) with each other per metric. To reduce the false discovery rate, the Benjamini-Hochberg procedure [64] was applied to the results for each metric. Aside from a few exceptions, all differences were significant, with adjusted p-values being below 0.05. The exceptions are the following:

- The original and the V3-encrypted event streams for the PSNR metric, which both result in infinite values for all



(a) original (b) V1 (c) V2 (d) V3 (e) V4 (f) decrypted

Figure 8: First reconstructed image of the scene ”boxes” for the original (a) event stream, the event streams encrypted using the different versions of the encryption algorithm (b)-(e), and the decrypted event stream (f).

scenes.

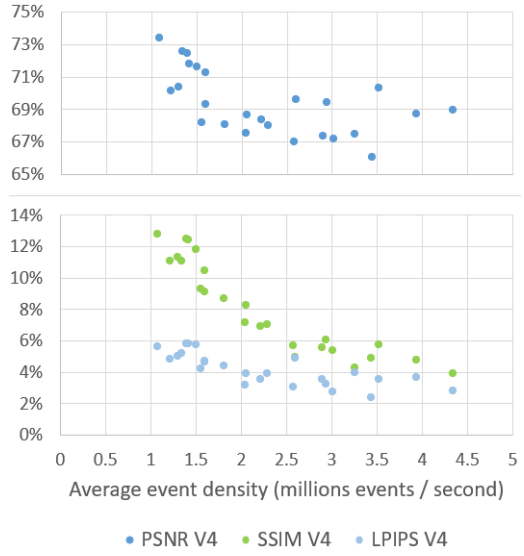
- The V3-encrypted and decrypted event streams for all metrics except PSNR.
- The original event stream with both the V3-encrypted and the decrypted event streams for the MSE metric.
- The V4-encrypted event stream with both the V3-encrypted and the decrypted event streams for the SemSim metric.

The data shows that encryptions V1 and V2, which use polarity mapping, outperform encryptions V3 and V4, which do not, in every metric. Moreover, while V4 outperforms V3 in the PSNR metric by a large margin, this large difference is not evident in the other metrics. However, the difference between V4 and V3 remains significant across SSIM, MSE, and LPIPS. While the differences are small, there are still significant differences between the encryptions V1 and V2. V1 outperforms V2 on PSNR and MSE, which compare individual pixels. On the other hand, V2 outperforms V1 on SSIM and LPIPS, which assess larger-scale structures or features in images.

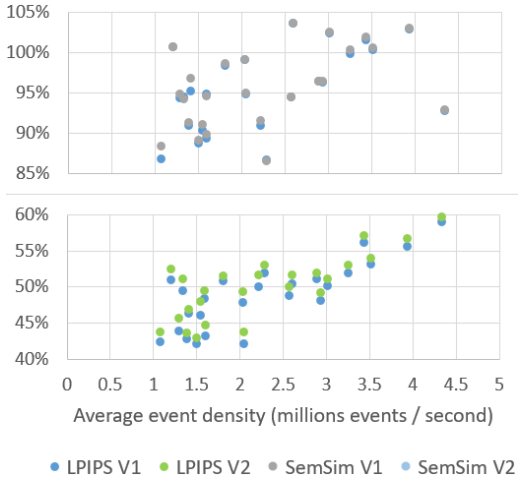
It should be noted that the SemSim measurement on the decrypted event stream yields a high score compared to the original event stream, contrary to the other metrics. This is unexpected, since the decrypted event streams retain a large amount of information, because they perform equally well in localisation as the original event streams. Moreover, SemSim claims to mimic human perception [53], but as shown in Figures 8a and 8f, the reconstructed images from the two streams look almost identical. A possible explanation is that I did not use the original model suggested by the authors, as discussed before.

Comparing with event characteristics

It has also been examined whether scene brightness, more specifically the average pixel brightness, or the motion of the event camera during capture, more specifically the average temporal event density, influences privacy preservation during encryption. This has been examined using regression analysis in Excel. For both pixel brightness and event density, the Benjamini-Hochberg procedure is applied to the results for all encryption variations of each metric to control the false discovery rate. The results are deemed significant if the adjusted p-value is below 0.05. According to the data, event density significantly affects the PSNR, SSIM, and LPIPS metrics for V4-encryption. The more events the camera captured per second, the worse the V4-encryption performed on those metrics, as shown in Figure 9a. On the other hand, as



(a) Significant relations between the event density and the metric scores for encryption V4.



(b) Significant relations between the event density and the metric scores for encryption V1 and V2.

Figure 9: Relative performance in different privacy metrics per scene depending on the temporal event density. Only the metrics that show a significant relation are shown. The 0% score is set at the score achieved by measuring the event reconstructions of the original event stream. The 100% score is set at the score achieved by measuring random noise.

shown in Figure 9b, higher event density yields better performance for V1- and V2-encryption across both LPIPS and SemSim. No significant relationship was found between average pixel brightness and privacy metrics.

6 Discussion

As mentioned in Chapter 5, the localisation performance is very likely to be positively related to the temporal event density. However, some event streams with relatively high event density have low accuracy scores, while others with low event density have high accuracy scores. This is probably because event density is not the only factor that differs between the event streams in the dataset. Other factors could also influence localisation performance, such as the scene brightness discussed earlier, as well as the scene size, the objects within the scene, and the path taken by the event camera during capture. While event density was used to measure the event camera's motion during capture, I do not think this result implies that faster movement leads to better localisation performance. I think a more likely explanation is that a higher movement speed allowed more passes over the same objects, resulting in more data to create the pre-built 3D map that the localisation algorithm compares the image to. However, event streams should first be captured under more controlled conditions to examine this more rigorously.

Since the encryption algorithm is the only factor that differs across the privacy measurements, comparing these encryptions allows a stronger conclusion to be drawn. The large difference in privacy performance between encryption variations with and without polarity mapping indicates that polarity mapping is responsible for most of the data obfuscation. While the differences are smaller, it is still interesting to see that the encryption algorithms that use Equation (4) to generate the timestamps, V2 and V4, perform better than the other algorithms. This is probably because these algorithms generate fewer noise events that bleed into the next chunk and are thus removed, resulting in more noise overall. In hindsight, the timestamp interval in Equation (4) over which noise events can be created remains too large, and more noise events could be generated if this interval is shortened.

The notable exceptions are the PSNR and MSE metrics for encryption algorithms that use polarity mapping. Of the two algorithms, the one that uses Equation (4) performs worse. A possible explanation is that when fewer noise events occur, image features are better preserved, resulting in lower privacy scores across the SSIM, LPIPS, and SemSim metrics. However, when polarity mapping is also applied, the pixel intensities might be flipped, creating larger differences than when all pixels converge to an average value due to noise. This results in higher scores on the pixel-based metrics PSNR and MSE. Lastly, it should be noted that the SemSim measurements were not acquired accurately, as mentioned before. If this data were to become available, these measurements could be performed correctly.

As described in Chapter 5, there are significant relationships between temporal event density and privacy metrics. As with the localisation performance, there are too many other factors to conclude with certainty that these relations are accurate. Nevertheless, I will still elaborate on my interpretation of the significant results. The relation shown in Figure 9a suggests that the larger the event density, the worse the V4 encryption performs on the PSNR, SSIM, and LPIPS metrics. A higher event density results in chunks with shorter time intervals. This leads to more noise events bleeding into succeeding chunks and being removed, resulting in poorer encryption. The relation shown in Figure 9b suggests that a larger event density corresponds to a better performance of the V1 and V2 encryptions for the LPIPS and SemSim metrics. This is odd, since we concluded that V2 outperforms V1 in LPIPS and SemSim, because more noise events are created, and that a higher event density leads to fewer noise events. In my opinion, this relation shows the dominant role the polarity mapping plays in the encryption algorithm. The privacy performance does not depend on the number of noise events, but on the event density itself. While the exact reason is unclear to me, I think E2VID struggles with temporally denser events that are partially flipped, and that its influence suppresses the influence of the decline in noise events. To make a stronger argument for these relations, more tests should be conducted where the relevant variables are isolated.

While I haven't explicitly measured it, I should mention that the encryption algorithm runs relatively slowly, and the generated key is relatively large, even after accounting for the fact that applying the Szudzik pairing function would reduce its size. These drawbacks should also be taken into account when assessing the algorithm's overall practicality.

7 Conclusions and Future Work

The research question this paper aims to answer is:

"How can the encryption of raw event camera data be practically and effectively used for privacy protection in a visual localisation application?"

This has been addressed by analysing the localisation and privacy performance of the localisation pipeline developed by Kim et al. [5] with different versions of the encryption and decryption algorithm developed by Zhang et al. [6]. We can conclude from the data that visual localisation works equally well on data that has been first encrypted and then decrypted as on the original data. Moreover, the dominant process obfuscating the event data is the polarity mapping. Any attacker could reverse the current polarity mapping process if they are aware that it is being used. Therefore, the encryption algorithm is not secure according to Kerckhoffs's principle [58]. Because of this, we conclude that the encryption and decryption algorithm designed by Zhang et al. can be used in practice in a visual localisation pipeline, but cannot guarantee effective privacy protection.

Further research should explore whether raw event encryption can be achieved using only polarity mapping, in which the mapping itself depends on a secret key. This would prevent attackers from reversing the process and would provide the privacy shown in this paper. Since no noise events would be created, the execution time can be decreased, and the key can be smaller. Furthermore, additional research could analyse the effects of scene characteristics on localisation and privacy performance in more controlled environments.

8 Responsible Research

Encryption of any privacy-sensitive data is important, as it enables users to securely use, store, and share their data without malicious actors obtaining it. As discussed before, event cameras are a relatively new technology, and little research has been done on encrypting the data they produce. In this paper, an algorithm for encrypting event camera data has been critically reviewed with respect to privacy preservation and practicality. If it were concluded that this algorithm is a huge success on both fronts, many other interested parties would implement the encryption and trust its privacy-preserving capabilities. However, if this were misjudged, many systems that use this algorithm would be vulnerable to attacks, and a lot of privacy-sensitive data would be stored and transferred insecurely. Therefore, it is important that the results are honest and accurate, and that the conclusion realistically follows from them and is not overly optimistic. Furthermore, gaps in the research must be addressed to avoid overlooking potential weaknesses, even if the results seem promising. For further review and research, I have shared my codebase, the dataset, and the hardware environment used to make the experiments easily reproducible.

A AI Statement

During this research project, artificial intelligence has been used for the following purposes:

- Suggesting code during implementation (Copilot).
- Reviewing and suggesting code improvements regarding code efficiency.
- Assisting in fixing bugs.
- Giving relevant context and possibly interesting papers to analyse regarding my research.
- Being a second opinion when questioning content like methodology and conclusions in other papers.
- Finding errors in spelling and grammar, and suggesting better formulations for my own paper (Grammarly).
- Reviewing my own paper and addressing gaps in the contents.

B Localisation Performance Results

Table 2: The localisation accuracy in percentages (*acc.*), the median translation error in meters (*t error*), and the median rotation error in degrees (*r error*) for all scenes. The respective pixel brightness (0-255) and temporal event density are also shown per scene.

			Original			Decrypted		
Scene	Avg. Bright.	Events/sec.	acc.	t error	r error	acc.	t error	r error
boxes	55.6	1.60E+06	81%	0.0389	1.3288	83%	0.0393	1.3531
cabinet	48.7	3.44E+06	99%	0.0206	1.1274	98%	0.0209	1.1810
cabinet_dark	28.7	1.81E+06	81%	0.0406	2.2341	82%	0.0389	2.0836
desk	108.8	3.94E+06	94%	0.0337	1.9258	94%	0.0343	1.8835
desk_dark	7.9	3.52E+06	98%	0.0343	1.7041	96%	0.0344	1.7000
fan	65.5	2.94E+06	93%	0.0427	1.2150	93%	0.0425	1.1955
kitchen_1	38.7	1.55E+06	81%	0.0503	1.2240	81%	0.0507	1.2329
kitchen_2	37.3	1.21E+06	71%	0.0604	1.6433	70%	0.0621	1.6712
lounge	100.8	2.60E+06	48%	0.1020	1.9361	46%	0.1028	2.0088
office_301	105.3	3.25E+06	97%	0.0282	1.2027	97%	0.0285	1.2163
office_inmc	72.1	2.22E+06	76%	0.0578	0.9371	75%	0.0543	0.9134
office_inmc_dark	47.9	1.50E+06	16%	0.2005	3.2774	17%	0.1999	3.1497
paris	45.5	1.39E+06	31%	0.1365	2.4154	34%	0.1335	2.2930
pingpong_1	44.2	1.29E+06	94%	0.0259	1.2405	95%	0.0254	1.1919
playroom	58.3	1.34E+06	89%	0.0385	1.0505	89%	0.0402	1.0936
pullup	52.1	2.05E+06	49%	0.1009	1.7621	51%	0.0981	1.7351
pullup_dark	40.1	1.08E+06	54%	0.0938	1.9029	54%	0.0938	1.9029
robot	65.8	2.58E+06	100%	0.0194	1.0546	100%	0.0190	1.0264
robot_dark	133.2	2.90E+06	88%	0.0408	2.0158	85%	0.0408	2.0704
room_1	39.5	2.04E+06	91%	0.0373	1.4320	90%	0.0383	1.4570
room_3	35.3	1.59E+06	76%	0.0551	2.0000	75%	0.0528	1.9614
sofa	51.9	3.02E+06	92%	0.0477	1.3185	92%	0.0441	1.2143
sofa_dark	38.2	1.42E+06	42%	0.1342	4.3515	39%	0.1452	3.8740
table	53.3	4.34E+06	100%	0.0132	0.8740	100%	0.0130	0.8609
table_dark	0.9	2.29E+06	94%	0.0242	1.2802	92%	0.0242	1.3069

C Privacy Metrics Results

C.1 PSNR

Table 3: The PSNR results, capped at 70, for all scenes for all types of images.

Scene	Original	Noise	V1	V2	V3	V4	Decrypted
boxes	70	8.8667	9.6522	10.2192	70	27.6297	60.8624
cabinet	70	8.352	10.5487	10.6823	70	29.2691	70
cabinet_dark	70	8.1891	9.5045	10.0147	70	27.9248	70
desk	70	8.1926	8.6658	8.7147	70	27.5382	58.0183
desk_dark	70	8.0899	9.2671	9.4909	70	26.4749	70
fan	70	8.975	10.9883	11.1922	70	27.6414	58.0915
kitchen_1	70	8.7535	10.4967	10.9512	70	28.2273	70
kitchen_2	70	8.151	8.4255	9.0323	70	26.6267	70
lounge	70	8.9068	10.037	10.1529	70	27.4764	70
office_301	70	8.3146	9.9474	10.0225	70	28.3719	59.3691
office_inmc	70	9.0192	12.4886	12.7281	70	28.2988	59.3766
office_inmc_dark	70	8.7433	10.9317	11.9901	70	26.1139	70
paris	70	8.506	10.731	11.7793	70	25.4507	70
pingpong_1	70	9.1609	11.5445	12.194	70	27.1974	70
playroom	70	8.8785	10.785	11.4048	70	25.6485	60.037
pullup	70	9.0099	11.7971	12.0705	70	28.1386	70
pullup_dark	70	8.8821	10.6435	11.3689	70	25.1526	60.2951
robot	70	8.6211	11.8867	12.0499	70	28.8993	70
robot_dark	70	8.1484	8.8774	9.0781	70	28.3274	58.7403
room_1	70	8.335	9.4568	9.6451	70	28.36	59.7256
room_3	70	8.9815	11.0745	11.5864	70	26.5006	70
sofa	70	8.3883	10.3138	10.4533	70	28.6238	70
sofa_dark	70	8.4927	9.8304	10.874	70	25.8315	59.4999
table	70	9.1979	11.9952	12.1175	70	28.0678	70
table_dark	70	8.4062	9.0464	9.2342	70	28.1098	59.205
Average	70	8.622488	10.357432	10.761884	70	27.436044	65.328832

Table 4: The p-values and adjusted p-values in decreasing significance for the paired two-tailed t-test on the relationships between the PSNR results of the different types of images.

Relationship	p-value	Adjusted p-value
Ori-Noise	1.57611E-55	0.0000
Noise-V3	1.57611E-55	1.65492E-54
Ori-V1	2.3462E-43	1.23175E-42
V1-V3	2.3462E-43	1.23175E-42
Ori-V2	1.05308E-42	3.6858E-42
V2-V3	1.05308E-42	3.6858E-42
Ori-V4	2.18274E-39	5.7297E-39
V3-V4	2.18274E-39	5.7297E-39
Noise-V4	4.68881E-30	1.09405E-29
V1-V4	6.92872E-27	1.45503E-26
V1-Dec	2.4868E-26	4.19825E-26
Noise-Dec	2.56149E-26	4.19825E-26
V2-Dec	2.59892E-26	4.19825E-26
V2-V4	2.8229E-25	4.23435E-25
V4-Dec	5.66678E-22	7.9335E-22
Noise-V2	1.84827E-11	2.42585E-11
Noise-V1	2.99847E-10	3.70399E-10
V1-V2	1.32048E-06	1.54056E-06
Ori-Dec	0.000234663	0.000246396
V3-Dec	0.000234663	0.000246396
Ori-V3	inf	inf

C.2 SSIM

Table 5: The SSIM results for all scenes for all types of images.

Scene	Original	Noise	V1	V2	V3	V4	Decrypted
boxes	1	0.009	0.3267	0.2985	0.9991	0.9099	0.9995
cabinet	1	0.0083	0.2468	0.2346	0.9991	0.9516	0.9994
cabinet_dark	1	0.0081	0.2479	0.2331	0.9989	0.9137	0.9995
desk	1	0.0082	0.2918	0.2797	0.9995	0.9528	0.9992
desk_dark	1	0.0081	0.2729	0.2627	0.9991	0.9427	0.9992
fan	1	0.0092	0.3206	0.3071	0.9997	0.9398	0.9994
kitchen_1	1	0.0089	0.3517	0.323	0.9998	0.9075	0.9996
kitchen_2	1	0.008	0.25	0.2278	0.9999	0.8899	0.9996
lounge	1	0.0093	0.4056	0.3894	0.9998	0.951	0.9995
office_301	1	0.0086	0.3617	0.3499	0.9991	0.9573	0.9994
office_inmc	1	0.0093	0.3679	0.3494	0.9998	0.9317	0.9995
office_inmc_dark	1	0.0091	0.3309	0.3098	0.9993	0.8828	0.9995
paris	1	0.0088	0.3281	0.3066	0.9998	0.8762	0.9995
pingpong_1	1	0.0094	0.3647	0.3283	0.9985	0.8879	0.9996
playroom	1	0.0091	0.2997	0.2735	0.9999	0.8899	0.9996
pullup	1	0.0093	0.3713	0.3441	0.999	0.9179	0.9995
pullup_dark	1	0.0089	0.3241	0.2924	0.9998	0.8733	0.9996
robot	1	0.0089	0.3756	0.362	0.9996	0.9434	0.9995
robot_dark	1	0.0081	0.2846	0.2712	0.9998	0.9446	0.9993
room_1	1	0.0083	0.2916	0.2704	0.9998	0.9291	0.9995
room_3	1	0.0092	0.3221	0.297	0.9992	0.8962	0.9996
sofa	1	0.0085	0.3025	0.2901	0.9995	0.9463	0.9994
sofa_dark	1	0.0086	0.2751	0.2595	0.9981	0.8765	0.9995
table	1	0.0095	0.3205	0.3119	0.9994	0.9611	0.9993
table_dark	1	0.0084	0.2699	0.251	0.999	0.9305	0.9994
Average	1	0.008764	0.316172	0.29692	0.99938	0.920144	0.999464

Table 6: The p-values and adjusted p-values in decreasing significance for the paired two-tailed t-test on the relationships between the SSIM results of the different types of images.

Relationship	p-value	Adjusted p-value
Noise-Dec	4.13736E-82	8.68845E-81
Ori-Noise	2.23187E-81	2.34346E-80
Noise-V3	6.47305E-78	4.53114E-77
Noise-V4	1.28418E-37	6.74192E-37
V2-V3	4.17016E-31	1.28953E-30
Ori-V2	4.26528E-31	1.28953E-30
V2-Dec	4.29845E-31	1.28953E-30
V1-V3	1.95002E-30	4.17216E-30
Ori-V1	1.98591E-30	4.17216E-30
V1-Dec	1.98675E-30	4.17216E-30
V2-V4	7.6124E-29	1.45328E-28
V1-V4	1.57143E-27	2.75E-27
Noise-V1	2.94613E-22	4.75914E-22
Noise-V2	5.73567E-22	8.60351E-22
Ori-Dec	1.10337E-17	1.54471E-17
Ori-V4	7.28068E-13	9.5559E-13
V3-V4	8.22337E-13	1.01583E-12
V4-Dec	8.95396E-13	1.04463E-12
V1-V2	2.5426E-12	2.81025E-12
Ori-V3	7.31755E-07	7.68343E-07
V3-Dec	0.384428938	0.384428938

C.3 MSE

Table 7: The MSE results for all scenes for all types of images.

Scene	Original	Noise	V1	V2	V3	V4	Decrypted
boxes	0	0.1304	0.1128	0.0999	0	0.0023	0
cabinet	0	0.1471	0.0939	0.0912	0.0001	0.0015	0
cabinet_dark	0	0.1526	0.1182	0.106	0	0.0021	0
desk	0	0.1521	0.1404	0.139	0	0.0025	0
desk_dark	0	0.1557	0.1242	0.1188	0	0.0032	0
fan	0	0.127	0.0839	0.0807	0	0.0028	0
kitchen_1	0	0.1337	0.0944	0.0858	0	0.0019	0
kitchen_2	0	0.1537	0.148	0.1294	0	0.0027	0
lounge	0	0.129	0.1023	0.0998	0	0.0023	0
office_301	0	0.148	0.1053	0.1035	0.0001	0.002	0
office_inmc	0	0.1258	0.0602	0.0567	0	0.0018	0
office_inmc_dark	0	0.1343	0.0862	0.0695	0	0.0031	0
paris	0	0.1416	0.0911	0.0724	0	0.0036	0
pingpong_1	0	0.122	0.0743	0.0642	0	0.0023	0
playroom	0	0.1298	0.0877	0.0765	0	0.0034	0
pullup	0	0.1265	0.0731	0.069	0	0.0019	0
pullup_dark	0	0.1301	0.0901	0.0773	0	0.0041	0
robot	0	0.1382	0.0715	0.0692	0	0.0017	0
robot_dark	0	0.1537	0.1337	0.1276	0	0.0023	0
room_1	0	0.1478	0.1189	0.114	0	0.0019	0
room_3	0	0.1268	0.0839	0.0744	0	0.0028	0
sofa	0	0.1455	0.097	0.0942	0	0.0018	0
sofa_dark	0	0.1422	0.1087	0.086	0.0001	0.0033	0
table	0	0.1205	0.0664	0.0646	0	0.0022	0
table_dark	0	0.1452	0.1313	0.1262	0	0.002	0
Average	0	0.138372	0.0999	0.091836	0.000012	0.00246	0

Table 8: The p-values and adjusted p-values in decreasing significance for the paired two-tailed t-test on the relationships between the MSE results of the different types of images.

Relationship	p-value	Adjusted p-value
Noise-V3	5.9355E-28	4.21911E-27
Ori-Noise	6.02729E-28	4.21911E-27
Noise-Dec	6.02729E-28	4.21911E-27
Noise-V4	1.07552E-27	5.64646E-27
Ori-V1	6.12573E-17	1.84027E-16
V1-Dec	6.12573E-17	1.84027E-16
V1-V3	6.13425E-17	1.84027E-16
V1-V4	1.04494E-16	2.74297E-16
Ori-V2	3.38087E-16	6.46797E-16
V2-Dec	3.38087E-16	6.46797E-16
V2-V3	3.38798E-16	6.46797E-16
V2-V4	6.71876E-16	1.17578E-15
Ori-V4	1.35559E-15	2.03338E-15
V4-Dec	1.35559E-15	2.03338E-15
V3-V4	1.75085E-15	2.4512E-15
Noise-V2	2.0605E-13	2.70441E-13
Noise-V1	1.33488E-11	1.64897E-11
V1-V2	1.02877E-06	1.20024E-06
Ori-V3	0.082985261	0.087134524
V3-Dec	0.082985261	0.087134524
Ori-Dec	1000	1000

C.4 LPIPS

Table 9: The LPIPS results for all scenes for all types of images.

Scene	Original	Noise	V1	V2	V3	V4	Decrypted
boxes	0	0.7767	0.3356	0.3468	0.0004	0.0361	0.0003
cabinet	0	0.8126	0.4556	0.4633	0.0003	0.0193	0.0003
cabinet_dark	0	0.8088	0.4108	0.4167	0.0005	0.0355	0.0003
desk	0	0.9326	0.5187	0.5287	0.0002	0.0345	0.0003
desk_dark	0	0.8789	0.4666	0.4741	0.0004	0.0313	0.0003
fan	0	0.777	0.3738	0.3815	0.0001	0.025	0.0003
kitchen_1	0	0.7482	0.3448	0.3588	0.0001	0.0316	0.0002
kitchen_2	0	0.7814	0.398	0.4093	0.0001	0.0378	0.0002
lounge	0	0.9139	0.4605	0.4715	0.0001	0.0445	0.0002
office_301	0	0.912	0.4732	0.4826	0.0005	0.0364	0.0002
office_inmc	0	0.7619	0.3808	0.393	0.0001	0.0272	0.0002
office_inmc_dark	0	0.7827	0.3289	0.3352	0.0003	0.0452	0.0003
paris	0	0.7691	0.3284	0.3346	0.0001	0.0447	0.0003
pingpong_1	0	0.7276	0.3189	0.3321	0.0005	0.0367	0.0002
playroom	0	0.7349	0.363	0.3751	0.0001	0.0381	0.0002
pullup	0	0.7426	0.3124	0.3244	0.0004	0.0293	0.0002
pullup_dark	0	0.7586	0.3208	0.3312	0.0001	0.0425	0.0002
robot	0	0.7838	0.3815	0.3917	0.0002	0.0238	0.0002
robot_dark	0	0.877	0.4476	0.4552	0.0001	0.031	0.0003
room_1	0	0.7749	0.3701	0.3818	0.0001	0.0245	0.0002
room_3	0	0.7595	0.3667	0.3749	0.0003	0.036	0.0002
sofa	0	0.7983	0.3996	0.408	0.0002	0.0218	0.0003
sofa_dark	0	0.7827	0.362	0.3665	0.0007	0.0454	0.0003
table	0	0.8651	0.5096	0.5162	0.0003	0.0246	0.0002
table_dark	0	0.8138	0.4226	0.4311	0.0005	0.032	0.0003
Average	0	0.802984	0.39402	0.403372	0.000268	0.033392	0.000248

Table 10: The p-values and adjusted p-values in decreasing significance for the paired two-tailed t-test on the relationships between the LPIPS results of the different types of images.

Relationship	p-value	Adjusted p-value
Noise-V1	6.54712E-30	1.37489E-28
Noise-V2	1.36475E-29	1.43298E-28
Ori-Noise	5.93805E-29	2.50215E-28
Noise-Dec	5.95242E-29	2.50215E-28
Noise-V3	5.9575E-29	2.50215E-28
Noise-V4	2.07548E-28	7.26417E-28
Ori-V2	1.13898E-21	2.69605E-21
V2-Dec	1.15242E-21	2.69605E-21
V2-V3	1.15545E-21	2.69605E-21
Ori-V1	2.54154E-21	4.50899E-21
V1-Dec	2.57111E-21	4.50899E-21
V1-V3	2.57656E-21	4.50899E-21
V2-V4	2.45563E-20	3.96678E-20
V1-V4	5.53683E-20	8.30525E-20
Ori-Dec	2.03259E-18	2.84563E-18
Ori-V4	2.80342E-17	3.67949E-17
V3-V4	3.11459E-17	3.84744E-17
V4-Dec	3.3204E-17	3.8738E-17
V1-V2	1.09215E-15	1.20711E-15
Ori-V3	8.50042E-08	8.92544E-08
V3-Dec	0.563650611	0.563650611

C.5 SemSim

Table 11: The SemSim results for all scenes for all types of images.

Scene	Original	Noise	V1	V2	V3	V4	Decrypted
boxes	0	19.272	18.2791	18.2141	13.5871	13.8475	13.4679
cabinet	0	20.1345	20.4312	20.5208	16.8157	16.8341	16.7064
cabinet_dark	0	18.8991	18.5928	18.6212	15.3987	15.6186	15.71595
desk	0	21.025	21.6216	21.6442	14.6002	14.6603	14.707625
desk_dark	0	20.8466	20.9037	20.9485	14.7658	14.9864	14.856125
fan	0	20.0764	19.3366	19.3544	15.4827	15.47	15.614525
kitchen_1	0	19.5392	17.6335	17.7754	14.0831	14.2786	14.337
kitchen_2	0	21.3714	21.5024	21.5036	15.2438	15.2516	15.41595
lounge	0	21.4487	22.2118	22.2287	16.1551	16.1326	16.43225
office_301	0	20.5737	20.5342	20.6231	16.7752	16.7997	16.665875
office_inmc	0	18.1997	16.5438	16.6501	14.2653	14.4166	14.286025
office_inmc_dark	0	18.589	16.4884	16.5532	14.6296	14.6468	14.593775
paris	0	18.0072	16.3727	16.4385	14.0977	14.5976	14.25325
pingpong_1	0	19.7611	18.6345	18.7302	16.0722	16.2725	16.271675
playroom	0	19.1947	18.1338	18.0901	14.9106	15.0906	14.799025
pullup	0	19.5738	18.5614	18.587	14.9605	15.0799	15.1148
pullup_dark	0	19.0913	16.5545	16.8622	13.5868	13.2304	13.18795
robot	0	19.7369	18.6494	18.6382	15.9303	16.0947	16.14855
robot_dark	0	19.4852	18.7771	18.7786	15.1259	15.4173	15.0662
room_1	0	19.2972	19.1102	19.1132	15.4996	15.5194	15.5852
room_3	0	19.5021	17.4205	17.5083	16.5869	16.6725	16.5679
sofa	0	19.8897	20.354	20.3798	15.9074	16.0887	15.863
sofa_dark	0	18.7513	17.8366	18.1448	15.8263	15.5293	15.771825
table	0	19.1983	17.8039	17.8219	14.8698	14.7053	14.795575
table_dark	0	18.7739	16.2703	16.2363	15.0101	14.7209	14.88885
Average	0	19.60952	18.74232	18.798656	15.207456	15.278476	15.244528

Table 12: The p-values and adjusted p-values in decreasing significance for the paired two-tailed t-test on the relationships between the SemSim results of the different types of images.

Relationship	p-value	Adjusted p-value
Ori-Noise	1.00529E-33	2.11111E-32
Ori-V4	4.16612E-31	3.23911E-30
Ori-V3	4.62729E-31	3.23911E-30
Ori-Dec	1.27363E-30	6.68658E-30
Ori-V2	1.17735E-26	4.94487E-26
Ori-V1	1.7385E-26	6.08474E-26
Noise-V3	1.98651E-17	5.95954E-17
Noise-Dec	2.30497E-17	6.05054E-17
Noise-V4	3.53934E-17	8.25845E-17
V2-Dec	1.63997E-11	3.44393E-11
V2-V3	2.11561E-11	4.03889E-11
V2-V4	2.66605E-11	4.66558E-11
V1-Dec	3.05699E-11	4.93821E-11
V1-V3	4.02208E-11	6.03312E-11
V1-V4	4.789E-11	6.7046E-11
Noise-V1	0.00015671	0.000205681
Noise-V2	0.000253707	0.000313402
V1-V2	0.004700011	0.005483347
V3-V4	0.082336614	0.091003626
V3-Dec	0.273830343	0.28752186
V4-Dec	0.371392635	0.371392635

D Relationships between Scene Characteristics and Privacy Metrics

Table 13: The p-values and adjusted p-values in decreasing significance for the regression analyses on the influence of the average temporal event density on the different privacy metrics performed on a certain image type.

Metric & Type	p-value	Adjusted p-value
SSIM-V4	2.44767E-10	4.89535E-09
LPIPS-V1	5.29749E-08	4.03779E-07
LPIPS-V2	6.05669E-08	4.03779E-07
LPIPS-V4	0.002276519	0.011382594
PSNR-V4	0.003760784	0.015043137
SemSim-V1	0.013282686	0.043619704
SemSIm-V2	0.015266896	0.043619704
MSE-V4	0.040595316	0.101488291
MSE-V2	0.187366716	0.359779731
SemSim-V3	0.194400415	0.359779731
PSNR-V2	0.206723527	0.359779731
SemSim-V4	0.215867839	0.359779731
MSE-V3	0.40752696	0.626964553
SSIM-V1	0.715589766	0.946461865
MSE-V1	0.74188055	0.946461865
SSIM-V2	0.762475919	0.946461865
PSNR-V1	0.804492585	0.946461865
SSIM-V3	0.871060918	0.967845464
LPIPS-V3	0.973547507	1.024786849
PSNR-V3	inf	inf

Table 14: The p-values and adjusted p-values in decreasing significance for the regression analyses on the influence of the average pixel brightness on the different privacy metrics performed on a certain image type.

Metric & Type	p-value	Adjusted p-value
SSIM-V4	0.028944876	0.304376399
SSIM-V2	0.034307773	0.304376399
SemSim-V1	0.068279073	0.304376399
SemSIm-V2	0.070378388	0.304376399
SSIM-V1	0.0760941	0.304376399
SSIM-V3	0.113825883	0.379419611
PSNR-V4	0.191132279	0.546092226
LPIPS-V2	0.310427078	0.694973379
LPIPS-V1	0.314195406	0.694973379
SemSim-V4	0.371823192	0.694973379
MSE-V4	0.382235359	0.694973379
SemSim-V3	0.490050026	0.755597194
MSE-V2	0.527678314	0.755597194
PSNR-V2	0.528918036	0.755597194
MSE-V3	0.593002548	0.790670064
PSNR-V1	0.863590094	0.968323896
LPIPS-V3	0.870285709	0.968323896
MSE-V1	0.871491507	0.968323896
LPIPS-V4	0.923625788	0.972237672
PSNR-V3	inf	inf

References

[1] Guillermo Gallego et al. "Event-Based Vision: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.1 (2022). DOI: 10.48550/arXiv.1904.08405.

[2] Jianing Li et al. "Active Event-based Stereo Vision". In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025, pp. 971–981. DOI: 10.1109/CVPR52734.2025.00099.

[3] Haoyue Liu et al. "NER-Net+: Seeing Motion at Nighttime With an Event Camera". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47.6 (2025), pp. 4768–4786. DOI: 10.1109/TPAMI.2025.3545936.

[4] Shafiq Ahmad et al. "Event-driven Re-Id: A New Benchmark and Method Towards Privacy-Preserving Person Re-Identification". In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. 2022, pp. 459–468. DOI: 10.1109/WACVW54805.2022.00052.

[5] Junho Kim et al. "Privacy-Preserving Visual Localization With Event Cameras". In: *IEEE Transactions on Image Processing* 34 (2025), pp. 6215–6230. DOI: 10.1109/TIP.2025.3607640.

[6] Pei Zhang, Shuo Zhu, and Edmund Lam. "Event encryption: rethinking privacy exposure for neuromorphic imaging". In: *Neuromorphic Computing and Engineering* 4 (Jan. 2024). DOI: 10.1088/2634-4386/ad207b.

[7] Misha A. Mahowald and Carver Mead. "The Silicon Retina". In: *Scientific American* 264.5 (1991), pp. 76–83. ISSN: 00368733, 19467087. URL: http://www.jstor.org/stable/24936904 (visited on 06/02/2026).

[8] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. "A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor". In: *IEEE Journal of Solid-State Circuits* 43.2 (2008), pp. 566–576. DOI: 10.1109/JSSC.2007.914337.

[9] 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". In: *IEEE Journal of Solid-State Circuits* 46.1 (2011), pp. 259–275. DOI: 10.1109/JSSC.2010.2085952.

[10] Christian Brandli et al. "A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor". In: *IEEE Journal of Solid-State Circuits* 49.10 (2014), pp. 2333–2341. DOI: 10.1109/JSSC.2014.2342715.

[11] Bongki Son et al. "4.1 A 640×480 dynamic vision sensor with a 9µm pixel and 300Meps address-event representation". In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017, pp. 66–67. DOI: 10.1109/ISSCC.2017.7870263.

[12] Yongcheng Jing et al. "Turning Frequency to Resolution: Video Super-resolution via Event Cameras". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 7768–7777. DOI: 10.1109/CVPR46437.2021.00768.

[13] Zeyu Xiao and Xinchao Wang. "Event-based Video Super-Resolution via State Space Models". In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025, pp. 12564–12574. DOI: 10.1109/CVPR52734.2025.01172.

[14] Yanxiang Wang et al. "EV-Gait: Event-Based Robust Gait Recognition Using Dynamic Vision Sensors". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6351–6360. DOI: 10.1109/CVPR.2019.00652.

[15] Yanxiang Wang et al. "Event-Stream Representation for Human Gaits Identification Using Deep Neural Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2022), pp. 3436–3449. DOI: 10.1109/TPAMI.2021.3054886.

[16] Paul K. J. Park et al. "Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique". In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 1624–1628. DOI: 10.1109/ICIP.2016.7532633.

[17] Waseem Shariff et al. "Event Cameras in Automotive Sensing: A Review". In: *IEEE Access* 12 (2024), pp. 51275–51306. DOI: 10.1109/ACCESS.2024.3386032.

[18] Xiao Wang et al. *Object Detection using Event Camera: A MoE Heat Conduction based Detector and A New Benchmark Dataset*. 2024. arXiv: 2412.06647 [cs.CV]. URL: https://arxiv.org/abs/2412.06647.

[19] Nuo Chen et al. *Event-based Tiny Object Detection: A Benchmark Dataset and Baseline*. 2025. arXiv: 2506.23575 [cs.CV]. URL: https://arxiv.org/abs/2506.23575.

[20] Garrick Orchard et al. "HFirst: A Temporal Approach to Object Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.10 (Oct. 2015), pp. 2028–2040. ISSN: 1939-3539. DOI: 10.1109/tpami.2015.2392947. URL: http://dx.doi.org/10.1109/TPAMI.2015.2392947.

[21] Junho Kim et al. *N-ImageNet: Towards Robust, Fine-Grained Object Recognition with Event Cameras*. 2022. arXiv: 2112.01041 [cs.CV]. URL: https://arxiv.org/abs/2112.01041.

[22] Hanme Kim et al. "Simultaneous Mosaicing and Tracking with an Event Camera". In: *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014* (Jan. 2014). DOI: 10.5244/C.28.26.

[23] Javier Hidalgo-Carrio, Guillermo Gallego, and Davide Scaramuzza. "Event-aided Direct Sparse Odometry". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 5771–5780. DOI: 10.1109/cvpr52688.2022.00569.

URL: http://dx.doi.org/10.1109/CVPR52688.2022.00569.

[24] Hanme Kim, Stefan Leutenegger, and Andrew Davison. "Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera". In: vol. 9910. Oct. 2016, pp. 349–364. ISBN: 978-3-319-46465-7. DOI: 10.1007/978-3-319-46466-4_21.

[25] Simon Klenk et al. *Deep Event Visual Odometry*. 2023. arXiv: 2312.09800 [cs.CV]. URL: https://arxiv.org/abs/2312.09800.

[26] Paul-Edouard Sarlin et al. *From Coarse to Fine: Robust Hierarchical Localization at Large Scale*. 2019. arXiv: 1812.03506 [cs.CV]. URL: https://arxiv.org/abs/1812.03506.

[27] Guillermo Gallego et al. "Event-Based, 6-DOF Camera Tracking from Photometric Depth Maps". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.10 (2018), pp. 2402–2412. DOI: 10.1109/TPAMI.2017.2769655.

[28] Samuel Bryner et al. "Event-based, Direct Camera Tracking from a Photometric 3D Map using Nonlinear Optimization". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 325–331. DOI: 10.1109/ICRA.2019.8794255.

[29] Anh Nguyen et al. *Real-Time 6DOF Pose Relocalization for Event Cameras with Stacked Spatial LSTM Networks*. 2018. arXiv: 1708.09011 [cs.CV]. URL: https://arxiv.org/abs/1708.09011.

[30] Hu Lin et al. "6-DoF Pose Relocalization for Event Cameras With Entropy Frame and Attention Networks". In: *Proceedings of the 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VRCAI '22. Guangzhou, China: Association for Computing Machinery, 2023. ISBN: 9798400700316. DOI: 10.1145/3574131.3574457. URL: https://doi.org/10.1145/3574131.3574457.

[31] Alex Kendall, Matthew Grimes, and Roberto Cipolla. *PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization*. 2016. arXiv: 1505.07427 [cs.CV]. URL: https://arxiv.org/abs/1505.07427.

[32] Gregory Cohen et al. "Spatial and Temporal Downsampling in Event-Based Visual Classification". In: *IEEE Transactions on Neural Networks and Learning Systems* 29.10 (2018), pp. 5030–5044. DOI: 10.1109/TNNLS.2017.2785272.

[33] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. *Continuous-time Intensity Estimation Using Event Cameras*. 2018. arXiv: 1811.00386 [cs.CV]. URL: https://arxiv.org/abs/1811.00386.

[34] Zelin Zhang, Anthony Yezzi, and Guillermo Gallego. "Formulating Event-based Image Reconstruction as a Linear Inverse Problem with Deep Regularization using Optical Flow". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 1–18. ISSN: 1939-3539. DOI: 10.1109/tpami.2022.3230727. URL: http://dx.doi.org/10.1109/TPAMI.2022.3230727.

[35] Zhi-Hong Guan, Fangjun Huang, and Wenjie Guan. "Chaos-based image encryption algorithm". In: *Physics Letters A* 346.1 (2005), pp. 153–157. ISSN: 0375-9601. DOI: https://doi.org/10.1016/j.physleta.2005.08.006. URL: https://www.sciencedirect.com/science/article/pii/S0375960105011904.

[36] S. Behnia et al. "A novel algorithm for image encryption based on mixture of chaotic maps". In: *Chaos, Solitons Fractals* 35.2 (2008), pp. 408–419. ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2006.05.011. URL: https://www.sciencedirect.com/science/article/pii/S0960077906004681.

[37] N.K. Pareek, Vinod Patidar, and K.K. Sud. "Image encryption using chaotic logistic map". In: *Image and Vision Computing* 24.9 (2006), pp. 926–934. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2006.02.021. URL: https://www.sciencedirect.com/science/article/pii/S026288560600103X.

[38] Xiaoling Huang. "Huang, X.L.: Image encryption algorithm using chaotic Chebyshev generator. Nonlinear Dyn. 67, 2411-2417". In: *Nonlinear Dynamics* 67 (Mar. 2012). DOI: 10.1007/s11071-011-0155-7.

[39] Lintian Qiao and Klara Nahrstedt. "A New Algorithm for MPEG Video Encryption". In: (Aug. 2001).

[40] Azfar Moid, Abraham Fapojuwo, and Robert Davies. "Secure and Scalable Video Streaming over IEEE 802.11e Based Home Networks". In: May 2007, pp. 445–448. ISBN: 1-4244-1020-7. DOI: 10.1109/CCECE.2007.118.

[41] Chung-Ping Wu and C.-C. Jay Kuo. "Efficient multimedia encryption via entropy codec design". In: *Security and Watermarking of Multimedia Contents III*. Ed. by Ping Wah Wong and Edward J. Delp III. Vol. 4314. International Society for Optics and Photonics. SPIE, 2001, pp. 128–138. DOI: 10.1117/12.435392. URL: https://doi.org/10.1117/12.435392.

[42] Bharath Ramesh and Hong Yang. "Boosted Kernelized Correlation Filters for Event-based Face Detection". In: *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. 2020, pp. 155–159. DOI: 10.1109/WACVW50321.2020.9096944.

[43] Federico Becattini, Federico Palai, and Alberto Del Bimbo. "Understanding Human Reactions Looking at Facial Microexpressions With an Event Camera". In: *IEEE Transactions on Industrial Informatics* 18.12 (2022), pp. 9112–9121. DOI: 10.1109/TII.2022.3195063.

[44] Henri Rebecq et al. *Events-to-Video: Bringing Modern Computer Vision to Event Cameras*. 2019. arXiv: 1904.08298 [cs.CV]. URL: https://arxiv.org/abs/1904.08298.

[45] Lin Wang et al. "Event-Based High Dynamic Range Image and Very High Frame Rate Video Generation Using Conditional Generative Adversarial Networks". In: *2019 IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR). 2019, pp. 10073–10082. DOI: 10.1109/CVPR.2019.01032.

[46] Binyi su, Lei Yu, and Wen Yang. "Event-Based High Frame-Rate Video Reconstruction With A Novel Cycle-Event Network". In: *2020 IEEE International Conference on Image Processing (ICIP)*. 2020, pp. 86–90. DOI: 10.1109/ICIP40778.2020.9191114.

[47] Lin Wang, Tae-Kyun Kim, and Kuk-Jin Yoon. *EventSR: From Asynchronous Events to Image Reconstruction, Restoration, and Super-Resolution via End-to-End Adversarial Learning*. 2020. arXiv: 2003.07640 [cs.CV]. URL: https://arxiv.org/abs/2003.07640.

[48] Mira Adra and Jean-Luc Dugelay. "E2PRIV: Privacy-Preserving Event-to-Video Reconstruction with Face Anonymization". In: *2025 13th International Workshop on Biometrics and Forensics (IWBF)*. 2025, pp. 1–6. DOI: 10.1109/IWBF63717.2025.11113401.

[49] Shafiq Ahmad, Pietro Morerio, and Alessio Del Bue. "Event Anonymization: Privacy-Preserving Person Re-Identification and Pose Estimation in Event-Based Vision". In: *IEEE Access* 12 (2024), pp. 66964–66980. DOI: 10.1109/ACCESS.2024.3399539.

[50] Bowen Du et al. "Event Encryption for Neuromorphic Vision Sensors: Framework, Algorithm, and Evaluation". In: *Sensors* 21.13 (2021). ISSN: 1424-8220. DOI: 10.3390/s21134320. URL: https://www.mdpi.com/1424-8220/21/13/4320.

[51] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study". In: *Journal of Computer and Communications* 07 (Jan. 2019), pp. 8–18. DOI: 10.4236/jcc.2019.73002.

[52] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV]. URL: https://arxiv.org/abs/1801.03924.

[53] Xiaoxiao Sun et al. *Privacy Assessment on Reconstructed Images: Are Existing Evaluation Metrics Faithful to Human Perception?* 2023. arXiv: 2309.13038 [cs.CV]. URL: https://arxiv.org/abs/2309.13038.

[54] Elias Mueggler et al. "The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM". In: *The International Journal of Robotics Research* 36 (Nov. 2016). DOI: 10.1177/0278364917691115.

[55] iniVation. *Davis 346 Camera*. https://shop.inivation.com/products/davis346. Accessed: 2024-03-01. 2022.

[56] cp-algorithms contributors. *Manhattan Distance — Algorithms for Competitive Programming*. https://cp-algorithms.com/geometry/manhattan-distance.html. Accessed: 2026-06-04. 2026.

[57] Matthew Szudzik. *An Elegant Pairing Function*. http://szudzik.com/ElegantPairing.pdf. Presented at the Wolfram Science Conference 2006. 2006.

[58] *Kerckhoffs's principle - Wikipedia — en.wikipedia.org*. https://en.wikipedia.org/wiki/{K}erckhoffs%27s_principle. [Accessed 11-06-2026].

[59] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[60] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 2)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2. 2024.

[61] KeerthikaMsft. *T.TEST function — Microsoft Support*. URL: https://support.microsoft.com/en-us/excel/t-test-function.

[62] GeeksforGeeks. *Regression in Excel*. July 2025. URL: https://www.geeksforgeeks.org/machine-learning/regression-in-excel/.

[63] Guanping Xiao et al. "Nondeterministic Impact of CPU Multithreading on Training Deep Learning Systems". In: *2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE)*. 2021, pp. 557–568. DOI: 10.1109/ISSRE52982.2021.00063.

[64] George Pipis and George Pipis. *The Benjamini-Hochberg procedure (FDR) and P-Value Adjusted Explained — R-bloggers*. July 2023. URL: https://www.r-bloggers.com/2023/07/the-benjamini-hochberg-procedure-fdr-and-p-value-adjusted-explained/.