The Spatio-Temporal Poisson Point Process: A Simple Model for the Alignment of Event Camera Data

Cheng Gu
TU Berlin
c.gu@campus.tu-berlin.de

Erik Learned-Miller
UMass Amherst
elm@cs.umass.edu

Daniel Sheldon
UMass Amherst
sheldon@cs.umass.edu

Guillermo Gallego
TU Berlin & Einstein Center Digital Future
guillermo.gallego@tu-berlin.de

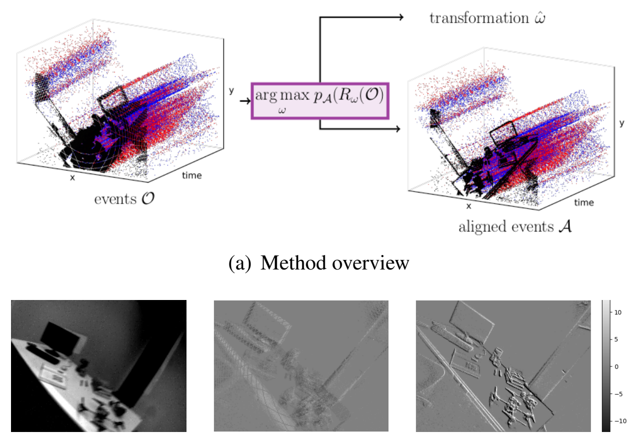
Pia Bideau
TU Berlin
p.bideau@tu-berlin.de

Abstract

Event cameras, inspired by biological vision systems, provide a natural and data efficient representation of visual information. Visual information is acquired in the form of events that are triggered by local brightness changes. However, because most brightness changes are triggered by relative motion of the camera and the scene, the events recorded at a single sensor location seldom correspond to the same world point. To extract meaningful information from event cameras, it is helpful to register events that were triggered by the same underlying world point. In this work we propose a new model of event data that captures its natural spatio-temporal structure. We start by developing a model for ***aligned*** *event data. That is, we develop a model for the data as though it has been perfectly registered already. In particular, we model the aligned data as a* ***spatio-temporal Poisson point process****. Based on this model, we develop a maximum likelihood approach to registering events that are not yet aligned. That is, we find transformations of the observed events that make them as likely as possible under our model. In particular we extract the camera rotation that leads to the best event alignment. We show new state of the art accuracy for rotational velocity estimation on the DAVIS 240C dataset [20]. In addition, our method is also faster and has lower computational complexity than several competing methods. Code:* https://github.com/pbideau/Event-ST-PPP

1. Introduction

Inspired by biological vision systems, event cameras [14, 25, 33, 4, 5] mimic certain biological features of the human vision system, such as recording brightness changes as *events*, asynchronously, and at high temporal resolution.



(a) Method overview

(b) Video frame

(c) Unaligned events, ‘blurred’ event image.

(d) Aligned events, sharp event image.

Figure 1. **Alignment of event data** by maximizing the joint probability of a set of events $p_{\mathcal{A}}(R_\omega(\mathcal{O}))$. Top row: Events are plotted in red/blue depending on their polarity. The projection of events onto the 2D image plane is shown in black - indicating the quality of their alignment over time. Aligned events projected onto 2D lead to sharp edge map, where as unaligned events are dispersed over the image plane. Bottom row: video frame, accumulated events, accumulated aligned events.

This relatively new way of acquiring visual information differs significantly from classical frame-based video recordings, leading to new research directions in computer vision and drawing close connections to robotics and the cognitive sciences. Prior work has shown that event data is rich enough to recover high quality brightness images, even in high-speed and high dynamic range (HDR) scenarios [26], and it allows early stage information processing such as motion perception and recognition [35, 12]. Despite these advantages, current vision algorithms still struggle to unlock the benefits of events cameras.

The problem of aligning event camera data. In this paper we focus on event camera data that comes from a moving camera in a static or nearly static environment. Because of the camera motion, as the camera records events through time, the events at a fixed camera pixel correspond to different points in the world. Conversely, many events recorded at different sensor pixel locations are corresponding to the same world point. This makes it more difficult to interpret event camera data. Finding transformations of the events that map each event triggered by the same world point to the same pixel location of the camera sensor can be called *alignment* or *registration* of the events. In this paper, we propose a method for alignment based on a new probabilistic model for event camera data.

Panoramas of events. To describe our model and algorithm, we draw analogies with *image panoramas* created using RGB images. By warping a set of images taken from different camera positions into the same shared set of coordinates, a set of images may be combined into a larger composite image, or panorama, of a scene.

The same idea can be applied to event data: transforming the location of each individual event so that it is transformed into a shared coordinate system [27, 11].[1] Doing this with event data is challenging, since it is more difficult to establish correspondences in event data than among images.

Instead, many approaches to registering event camera data are based upon a simple intuitive observation [19, 30, 15, 21, 7]. If we form an "aggregate" event camera image by simply recording the number of events at each pixel over some period of time, then these aggregate images tend to be sharper when the events are well-aligned (Figure 1(d)), and blurrier when the events are less well-aligned (Figure 1(c)). Leveraging this observation, one tries to find a set of transformations that maximize the sharpness of the aggregate image. These methods, discussed in detail in the related work section, mostly differ in their definition of defining sharpness, i.e., in their loss functions.

Congealing and probabilistic models of alignment. In this paper, we introduce a new, more effective method for event alignment. It is related to a probabilistic method for aligning traditional images known as *congealing* [13], which does not use any explicit correspondences. Instead, one measures the degree to which a set of images are *jointly aligned*. To measure the quality of the joint image alignment, one considers the *entropy* of the set of pixels at each image location. If a location has the same pixel value across all of the images, it has minimum entropy. If it has many different pixel values, it has high entropy. By transforming the images so that the sum of these pixelwise entropies is minimized, the images naturally move into alignment. Since minimizing entropies is equivalent to maximizing pixel likelihoods under a non-parametric distribution, congealing can also be seen as a *maximum likelihood method* (see [13] for more details).

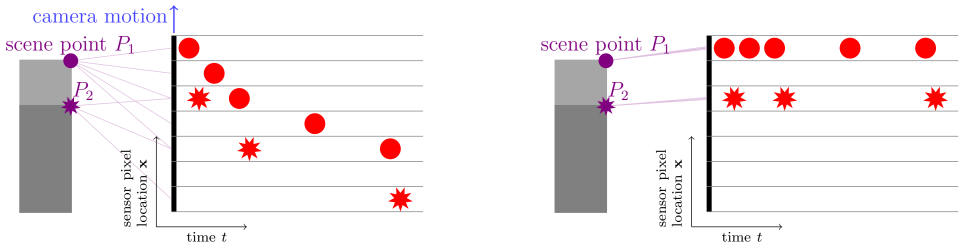
Contributions. We present a novel probabilistic model for event camera data. It allows us to evaluate the likelihood of the event data captured at a particular event camera pixel. By introducing transformations to move the data into a common coordinate system, we show that by maximizing the likelihood of the data under this model with respect to these transformations, we naturally retrieve an accurate registration of the event data. That is, we develop a probabilistic, maximum likelihood method for the joint alignment of event data. We support this novel approach by providing new state-of-the-art results. We have substantially higher accuracy than recently published methods, and are also among the fastest. In addition, we reassess how evaluations on these *de facto* benchmarks are done, and argue that a new approach is needed.

2. Related Work

Rotational velocity estimation from event data has been an active research topic since an Inertial Measurement Unit (IMU) was integrated on the Dynamic Vision Sensor (DVS) event camera [14] to yield a combined visual and vestibular device [3] (the precursor of the DAVIS240 event camera [1]). Sensor fusion between the IMU's gyroscope (measuring angular velocity) and the DVS output allowed the stabilization of events for a short amount of time. However, IMUs are interoceptive sensors that suffer from biases and drift (as error accumulates), so exteroceptive solutions using the events were investigated as alternative means to estimate rotational motion and therefore to stabilize event-based output during longer time periods.

Early work on rotational motion estimation (i.e., camera tracking) from event data includes [2, 11, 27, 8]. Some of these works arose as 3-DOF (degrees of freedom) Simultaneous Localization and Mapping solutions [11, 27]. Since depth cannot be estimated in purely rotational motions, the "mapping" part refers to the creation of a panoramic image (either of edges [27] or of reconstructed brightness [11]). The method in [8] proposed to estimate rotational motion by maximizing the contrast of an image of displaced (warped) events. This contrast measure was the highest quality metric for event alignment in terms of accuracy and computation time among 24 loss functions that were explored in [6]. The event alignment technique was later applied to other problems (depth, optical flow and homography estimation) in [7]. Since then, the idea of event alignment has been gaining popularity and extended, via different alignment metrics and optimization algorithms, for several motion estimation and segmentation problems

[1] In the event camera literature, the term 'panorama' is usually applied to alignment over sequences in which the camera has large displacements, resulting in a panorama much larger than a single camera frame. However, the same term can be applied to registering short sequences of event camera data, which creates panoramas only slightly larger than a single frame.



(a) **Observed events** (b) **Aligned events**

Figure 2. **The spatio-temporal Poisson point process.** Events arise due to the movement of the camera relative to the scene. Here, events are pictured in red. Events with the same shape are triggered by the same scene point. Each row indicates different sensor pixel locations over time. (a): As the camera moves, events capturing the same scene point in the world are recorded at different sensor locations - we call them 'de-registered'. (b): Events that can be associated with the same scene point in the world are registered to each other and are modeled as aligned Poisson point processes. *Ironically* aligned events are a useful representation of event data to extract scene information, but actually events are only triggered if the camera is moving. Thus event data can only be acquired in its de-registered form (a).

in [6, 19, 30, 31, 21, 15, 24, 23, 22].

The closest work to us are [21, 7]. In [21] event alignment is expressed via a family of entropy functions over *all pairs* of warped events. Entropy measures dispersion and our approach can also be interpreted as an entropy minimization [13]. In contrast, we propose a framework that maximizes the likelihood of events at *each pixel location*, as opposed to using pairwise event measures. This directly corresponds to minimizing the entropy *per pixel*, independently. Assuming pixel-wise independence allows to derive an event alignment approach that is computationally efficient (reduced complexity) and achieves high performance, as shown in the experiments (Section 4). In addition, independent modeling of each pixel leads to a simple theoretical formulation with clear properties and dependencies.

Congealing and probabilistic models for alignment. Our event alignment method is inspired by *congealing* [18] –a probabilistic approach for joint alignment of image data. Congealing aligns images by maximizing their joint probability under a set of transformations to make the images as similar as possible to each other. Congealing has been successfully applied to align binary images (e.g., MNIST), medical MRI volumes [36], complex real-world images like faces [9, 10] and 1D curves [17]. In this work, we further develop the principles of congealing to align the unconventional visual data produced by event cameras. The result is a new probabilistic approach that, while being developed for rotational motion estimation, also extends to related event alignment problems [7, 21, 23].

3. A probabilistic model for event data

In this section, we present our probabilistic model for event data. We start by defining two types of 'event processes'. These processes are models for the *observed* event data, which is unaligned, and data that has been perfectly aligned using ground truth transformations. These two processes are illustrated in Figure 2.

3.1. The observed data

The observed data is a set of N events recorded by a moving event camera over a time period ΔT. We denote the observed events as

$$\mathcal{O} = [\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_N], \tag{1}$$

where $\mathbf{o}_i = (o_i^{\mathbf{x}}, o_i^t)$ comprises the pixel location $o_i^{\mathbf{x}}$ and the time o_i^t at which the event occurred on the image plane.

3.1.1 The observed pixel processes

Consider the set of all events recorded at the *same pixel* location $\mathbf{x}$ in the event camera. Among the events $\mathcal{O}$, the subset of events $\mathcal{O}^{\mathbf{x}}$ that occur at a specific pixel is

$$\mathcal{O}^{\mathbf{x}} = \{\mathbf{o}_i : o_i^{\mathbf{x}} = \mathbf{x}\}. \tag{2}$$

We refer to such a set of observed events generated at a particular event camera location as an *observed pixel process*. Each row of events in Figure 2(a) shows such a process. The different shapes in each row illustrate that these pixel processes were generated by different scene points. However, to the camera, the events look the same, irrespective of what world point they were generated from.

We can define an observed pixel process for each of the N_P pixels in the event camera, resulting in a set of N_P observed pixel processes. We define an observed pixel process for a pixel even if there were no events observed at that pixel. That is, some observed pixel processes may not have any events associated with them.

3.2. The aligned data

Next, we consider the events as though they have been perfectly aligned with a set of ideal or ground truth transformations. We describe this as the set of events

$$\mathcal{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N].$$

Here, $\mathbf{a}_i = (a_i^{\mathbf{x}}, a_i^t)$ represents an event whose location has been transformed according to

$$a_i^{\mathbf{x}} = T_{GT}(o_i^{\mathbf{x}}; t)$$

where $T_{GT}(\cdot; t)$ is a *ground truth* transformation that exactly inverts the camera motion. We define this ground truth transformation function T_{GT} to be one that maps each event caused by a particular world point P to the same location in the pixel camera coordinate system.

This set of aligned events can be thought of as an event panorama in which all of the events have been registered.[2] That is, any events that emanated from the same world point should now have the same coordinates.

Note that like a traditional image panorama, the registration of points is likely to create an 'image' which is larger than the original camera image, since we are effectively overlaying a bunch of different images into the same coordinates. Thus, while events are being registered into the same coordinate frame, the actual coordinates may extend beyond the limits of the original image.

3.2.1 The aligned pixel processes

Now for the case of the aligned data, consider the set of all events with the same transformed pixel location $\mathbf{x}$. That is, among the events $\mathcal{A}$, we define the subset of events $\mathcal{A}^{\mathbf{x}}$ that occur at a specific pixel location $\mathbf{x}$:

$$\mathcal{A}^{\mathbf{x}} = \{\mathbf{a}_i : a_i^{\mathbf{x}} = \mathbf{x}\}.$$

We refer to such a set of aligned events at a particular location as the output of an *aligned pixel process*. By definition, each event $\mathbf{o}$ that originates at world point P is transformed to the same image point (with location rounded to the nearest pixel center) $a^{\mathbf{x}}$ by the ground truth transformation $a^{\mathbf{x}} = T_{GT}(o^{\mathbf{x}}; t)$. Collectively, the aligned data can be viewed as a set of aligned pixel processes.

3.3. A probabilistic model for aligned data

We now introduce our probabilistic model for *aligned event camera data* and describe how it can be used to align observed (i.e., unaligned) data.

First consider a model for a single aligned pixel process, representing all of the events associated with a particular pixel location $\mathbf{x}$. We model this as a *Poisson process* [32] with a rate parameter $\lambda_{\mathbf{x}}$. Henceforth, assume for simplicity that time is rescaled so the observation interval length is $\Delta T = 1$. This implies that the number of events $k_{\mathbf{x}} = |\mathcal{A}^{\mathbf{x}}|$ occurring at location $\mathbf{x}$ is a Poisson random variable with parameter $\lambda_{\mathbf{x}}$:

$$p(k_{\mathbf{x}}) = \text{Pois}(k_{\mathbf{x}}|\lambda_{\mathbf{x}}) = \frac{\lambda_{\mathbf{x}}^{k_{\mathbf{x}}} e^{-\lambda_{\mathbf{x}}}}{k_{\mathbf{x}}!}. \tag{3}$$

[2]If the camera motion contains translations, then this can only be done approximately.

Next, we model the entire aligned data set as the output of a collection of independent Poisson point processes, each with a separate rate $\lambda_{\mathbf{x}}$ depending upon its location. By standard properties of Poisson processes [32], this is equivalent to a *single* Poisson point process over space and time—i.e., a *spatio-temporal Poisson point process* (ST-PPP)—with intensity function $\lambda(\mathbf{x}, t) \doteq \lambda_{\mathbf{x}}$. By definition, the events of an ST-PPP at one spatial location are independent of those at other locations. Fig. 2(b) illustrates such an ST-PPP.

Let $\mathcal{X}$ be the set of locations at which events occur in the aligned event camera data. Then, due to independence over spatial locations, we can write the probability of the entire aligned data set under the ST-PPP model as[3]

$$p_{\mathcal{A}}(\mathcal{A}) \doteq \prod_{\mathbf{x}\in\mathcal{X}} \text{Pois}(k_{\mathbf{x}}|\lambda_{\mathbf{x}}), \quad k_{\mathbf{x}} = |\mathcal{A}^{\mathbf{x}}|. \tag{4}$$

3.4. An optimization problem

The model above leads naturally to an optimization problem. We shall seek a set of transformations, one applied to each event, that maximizes the likelihood of the transformed data under our model. Because the interval ΔT over which we are considering camera motions is very small (just a fraction of a second), we adopt the typical assumption that our transformations are smooth with respect to time. While we consider other families of transformations in the experiments section, we describe our optimization with respect to sets of *constant angular velocity* rotations:

$$R_\omega^t = \exp(S(\omega) \cdot t), \tag{5}$$

where $S(\omega)$ is a skew-symmetric matrix that encodes the 3-parameter angular velocity ω and whose exponentiation leads to a rotation matrix. Here t is the time of the recorded event, which is set to 0 for the beginning of the sequence and ΔT at the end of the sequence. Since t scales the angular velocity ω, it controls the amount of rotation, and hence the amount of rotation is a linear function of t.

To transform events, we define R_ω^t as the mapping $(\mathbf{x}, t) \mapsto (R_\omega^t \mathbf{x}, t)$ that applies the time-dependent rotation to the event location $\mathbf{x}$ and preserves the event time t. In this way, each event is rotated an amount proportional to the time at which it occurred.

To optimize the alignment of events over this set of choices for transformations, we solve for

$$\hat{\omega} = \arg\max_{\omega\in\Omega} p_{\mathcal{A}}\big(R_\omega(\mathcal{O})\big), \tag{6}$$

where $R_\omega(\mathcal{O}) = [R_\omega^{t_0}(\mathbf{o}_1), \ldots, R_\omega^{\Delta T}(\mathbf{o}_N)]$. Here, we have implicitly defined the likelihood $p_{\mathcal{O}}(\mathcal{O}|\omega)$ of the observed

[3]We slightly abuse notation with the notation $p_{\mathcal{A}}(\mathcal{A})$; our expression gives the probability of the counts $k_{\mathbf{x}}$, which differs from the density of the point set $\mathcal{A}$ by a factor of $k_{\mathbf{x}}!$.

data through the mapping $p_{\mathcal{O}}(\mathcal{O}|\omega) = p_{\mathcal{A}}\big(R_\omega(\mathcal{O})\big)$. This can be formally justified through the Poisson mapping theorem [32]. We give more background on this in the supplementary material.

The formula in (4) assumes knowledge of the Poisson rate parameter $\lambda_{\mathbf{x}}$ at each location $\mathbf{x}$. One option would be to estimate these parameters via maximum likelihood jointly with ω. Instead, we adopt a partially Bayesian approach by maximizing the marginal likelihood of $k_{\mathbf{x}}$ under the prior $\lambda_{\mathbf{x}} \sim \text{Gamma}(r, q^{-1}(1-q))$, for fixed parameters $r > 0$ and $q \in [0, 1]$. Then, by the well-known construction of the negative binomial distribution as a Gamma-Poisson mixture, the marginal distribution of $k_{\mathbf{x}}$ is $\text{NB}(r, q)$, which we can compute and optimize directly. Our final model for aligned data, which we will use in place of Eq. (4), is

$$p_{\mathcal{A}}(\mathcal{A}) = \prod_{\mathbf{x}\in\mathbf{X}} \text{NB}(k_{\mathbf{x}}|r, q), \quad k_{\mathbf{x}} = |\mathcal{A}^{\mathbf{x}}|. \tag{7}$$

We discuss approaches to estimate the parameters r and q in the experiments section.

3.4.1 Transformations

Another choice in event camera alignment algorithms is the choice of transformations. In the most general setting $T(\cdot)$ could be any smooth and invertible map from coordinates $(\mathbf{x}, t)$ to new coordinates $(\mathbf{x}', t)$ describing the new spatial-temporal location of events. Here we focus on camera rotations R_ω^t as the set of possible transformations, however other transformations such as translations and their combinations are possible. Possible extensions are discussed in the experiments section in further detail.

3.5. Implementation details

Event polarity. Until now, we have been considering a single uniform type of event, but most event cameras output either *positive* or *negative* events depending upon the sign of brightness changes. There are various ways to deal with the diversity of events. One option would be to treat all events as equivalent, irrespective of their polarity, but this would discard information. Instead, we treat positive and negative events as arising from *independent* ST-PPPs— in other words, the number of positive events $k_{\mathbf{x}}^+$ and negative events $k_{\mathbf{x}}^-$ at pixel $\mathbf{x}$ in the aligned process are independent Poisson random variables with rates $\lambda_{\mathbf{x}}^+$ and $\lambda_{\mathbf{x}}^-$, respectively. With $\lambda_{\mathbf{x}}^+, \lambda_{\mathbf{x}}^- \sim \text{Gamma}(r, q^{-1}(1-q))$, this gives the likelihood

$$p_{\mathcal{A}}(\mathcal{A}) = \prod_{\mathbf{x}\in\mathcal{X}} \text{NB}(k_{\mathbf{x}}^+; r, q) \cdot \text{NB}(k_{\mathbf{x}}^-; r, q). \tag{8}$$

Operationally, this corresponds to separately computing the log loss for positive and negative events and adding them together to get a total loss.

Optimization. We optimize the loss function (8) using the Adam algorithm implemented in the Python package torch.optim with a learning rate of 0.05 and a maximum number of iterations set to 250. No learning rate decay is applied. Similar to [7, 8] we sequentially process packets of $N = 30000$ events, and like [8] we smooth the image of warped (IWE) events using a Gaussian filter with a small standard deviation ($\sigma = 1$) making the algorithm less susceptible to noise. We apply a padding of 100 pixels, such that in most cases all recorded events originating from the same world point are aligned with each other and are taken into account for the computation of the loss function. The loss function is normalized by the number of events present on the image plane.

4. Experiments

We evaluate our approach on publicly available data [20]. We discuss the results and show an ablation study to support the understanding of our proposed approach for motion estimation from the output of an event camera. Our approach is based on the event data only and does not require any other additional information such as video frames.

4.1. Dataset and Evaluation Metrics

The **DAVIS 240C Dataset** [20] is the de facto standard to evaluate event camera motion estimation [8, 15, 21, 27, 34, 28]. Each sequence comprises an event stream, video frames, a calibration file, and IMU data from the camera as well as ground truth camera poses from a motion capture system. The gyroscope and accelerometer of the IMU output measurements at 1kHz. The motion capture system provides ground truth camera poses at 200Hz. The spatial resolution of the DAVIS camera [1] used is 240×180 pixels. The temporal resolution is in the range of microseconds. We evaluate our approach on sequences *boxes*, *poster*, *dynamic* and *shapes*. All sequences have 1 minute duration, 20–180 million events and an increasing camera motion over time.

Evaluation metrics. The dataset [20] does not come with an associated evaluation protocol. We therefore define an evaluation protocol in accordance to previous work for angular velocity estimation. Typically algorithms for angular velocity estimation estimate a constant velocity ω over a fixed set of N events. Let t_{start} be the time stamp of the first event within the set of N events and t_{end} be the time stamp of the last event. We compare the estimated velocity ω with the ground truth at time $t_{\text{mid}} = (t_{\text{end}} - t_{\text{start}})/2$. Similar to [21] we evaluate all methods using four different error measurements: angular velocity error $(e_{\omega_x}, e_{\omega_y}, e_{\omega_z})$ in degrees/s, their standard deviation σ_{e_ω}, the RMS-error in degrees/s. The RMS error compared to the maximum excursions of ground truth is presented as a percentage (%).

	Method	e_{wx}	e_{wy}	e_{wz}	σ_{ew}	RMS	RMS%
boxes	CMax [7]	7.38	6.66	6.03	9.04	9.08	0.66
	AEMin [21]	6.75	5.19	5.78	7.77	7.81	0.56
	EMin [21]	**6.55**	4.40	5.00	7.00	7.06	0.51
	Poisson Point-Proc.	6.72	**3.93**	**4.55**	**6.64**	**6.73**	**0.49**
poster	CMax [7]	13.45	9.87	5.56	13.39	13.45	0.74
	AEMin [21]	12.57	7.89	5.63	12.35	12.36	0.68
	EMin [21]	11.83	7.31	4.37	10.85	10.86	0.60
	Poisson Point-Proc.	**11.78**	**6.33**	**3.67**	**10.30**	**10.37**	**0.57**
dynamic	CMax [7]	4.93	4.82	4.95	7.11	7.13	0.71
	AEMin[21]	5.02	3.88	4.55	6.16	6.19	0.62
	EMin [21]	4.78	3.72	3.73	5.33	5.39	0.54
	Poisson Point-Proc.	**4.42**	**3.61**	**3.49**	**5.15**	**5.19**	**0.52**
shapes	CMax [7]	31.19	26.83	38.98	55.86	55.87	3.94
	AEMin [21]	22.22	18.78	35.41	55.43	55.44	3.91
	EMin [21]	21.22	15.87	25.57	42.22	42.22	2.98
	Poisson Point-Proc.	**20.73**	**13.95**	**17.69**	**25.88**	**25.89**	**1.83**

Table 1. **Angular velocity estimation**. Accuracy comparison on the rotation sequences from dataset [20].

We also show results on linear velocity estimation of the camera. Since the depth of the scene is not provided [20], linear velocity with a single camera can only be estimated up to scale. In this case we compare the estimated and ground truth linear velocities by computing a scale factor between them (via linear regression).

Ground truth. The built-in gyroscope of the DAVIS' IMU [3] provides accurate measurements of the camera's orientation and therefore is used in this paper to evaluate the camera's angular velocity. As reported by [20], measurements of the IMU come with a temporal lag of $\approx$ 2.4ms, which we take into account in our evaluation pipeline. On the other hand, the quality of the data produced by the DAVIS' IMU accelerometer does not match the high positional accuracy of the motion capture system. Hence we use the latter for linear velocity assessment.

4.2. Results

Angular velocity estimation. We compare our method (maximization of likelihood (8)) to the most recent work for angular velocity estimation [7, 21]. Gallego et al. [7, 6] estimate motion by maximizing the contrast (e.g., variance) of an image of warped events (IWE). Nunes et al. [21] estimate motion by minimization of an entropy (e.g. Tsallis') defined between pairs of events in the spatio-temporal volume. They provide an exact entropy calculation, which is expensive, and an approximate one, which is faster. We compare against both, in terms of accuracy and runtime.

Table 1 shows the quantitative comparison of accuracy among all event-based angular velocity estimation methods on all four rotational motion sequences. Our Poisson point process method consistently outperforms the baseline methods. On poster_rotation and boxes_rotation our approach shows an improvement of about 5% measured based on the root mean square error (RMS). On the other two sequences we show improvement of 4% and 39% compared to the next best performing method. The gain of 39% in particular shows our superior performance for event recordings where the scene structure (brightness) and motion varies significantly. The shapes sequence contains much fewer events due to its "simple" scene structure than the other three sequences of this dataset. Even during peak velocities of about ± 940 deg/s, which corresponds to 2.5 full rotations per second, our approach robustly estimates the motion.

Method	e_{vx}	e_{vy}	e_{vz}	σ_{ev}	RMS	RMS%
CMax [7]	0.21	0.26	0.41	0.42	0.43	7.83
AEMin [21]	0.21	0.25	0.42	0.44	0.45	8.36
EMin [21]	0.35	0.43	0.46	0.63	0.64	11.80
Poisson Point-Proc.	**0.17**	**0.22**	**0.38**	**0.38**	**0.38**	**6.93**

Table 2. **Linear velocity estimation.** Accuracy comparison on four translational sequences from dataset [20]. Average results.

Figure 3 shows qualitative results for all four approaches used for comparison. We show the ground truth aligned event image together with a detailed close-up view for each approach highlighting the alignment quality of events at object edges. Our approach consistently reconstructs very 'sharp' object contours (see first three rows of Fig. 3). The shapes sequence (depicted in the last column of Fig. 3) comes with quite different image characteristics. Due to its rather simple structure, this event sequence comprises roughly 20% of the amount of events that are usually acquired during the same time. Therefore a fixed number of events that is consistently used over all four data sequences results in much larger time intervals containing highly varied motion. In these cases none of the algorithms is able to align the events accurately assuming constant velocity during a fixed number of events.

Linear velocity estimation. Since our approach is flexible to the type of spatial transformation considered, we also assess its performance on the estimation of translational camera motion, e.g., linear velocity. Table 2 summarizes the results of event-based linear velocity estimation using also non-overlapping packets of 30k events. For this task, we use the same textured scenes in [20], but the set of sequences with translational motion. A challenge for all methods evaluated here is to avoid all events warping to a single pixel (undesired minima of the alignment measures), which can happen for large Z-motions.

Additional visual results for velocity estimation are provided in supplementary material.

Runtime and time complexity analysis. We measure the time that it takes to compute the alignment (loss) function given a set of 30k events. For comparability we reimplement our loss function in C++. The runtime was mea-

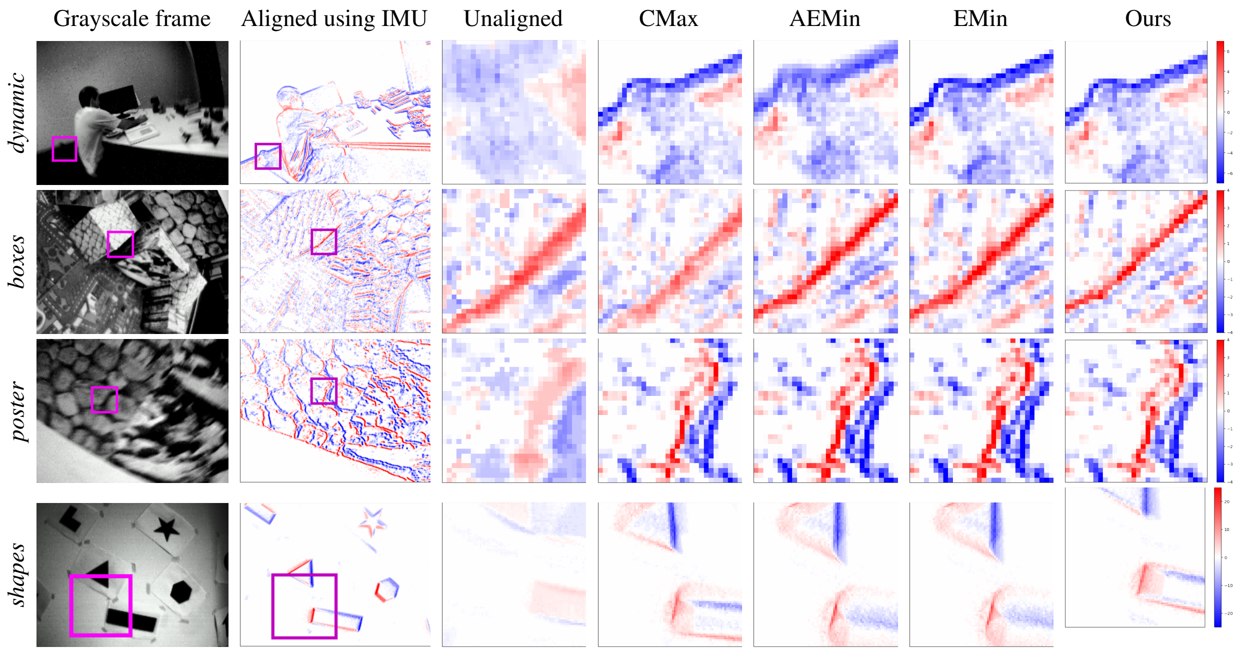


Figure 3. **Qualitative results** for each sequence with predominant rotational camera motion. From top to bottom: dynamic_rotation, boxes_rotation, poster_rotation and shapes_rotation. From left to right: Grayscale frame, aligned event image using ground truth from IMU, unaligned events, CMax, AEMin, EMin, Ours. All methods assume constant velocity for a batch with a fixed number of events. Note that the *shapes* sequence comprises a sparser scene texture, thus a batch of a fixed number of events spans over a larger time interval with mostly more variation in camera motion. In this sequence, accurate alignment is not possible for any of the methods under the constant velocity assumption. The affine model of angular velocity that we propose in Section 4.3 mitigates this issue as shown in Fig 6(f).

sured using an 8-core CPU with 16 threads and clock speed of 3.9 GHz. The runtime of all four tested methods is compared in Fig. 4, and plotted against accuracy. Our approach achieves highest average accuracy for angular velocity estimation and is among the fastest (3.1ms for one loss function evaluation). The contrast maximization approach is the fastest approach taking just 1.2ms per function evaluation, but comes with significantly lower performance in terms of the RMS measure.

Additionally, the complexity analysis in Table 3 explains the slow computation time of both entropy minimization methods (EMin, AEMin). The complexity of our approach as well as for contrast maximization is linear with the number of events N_e. The complexity of EMin [21] is quadratic with the number of events since it requires the evaluation of costs due to all pairs of events. The faster, approximate version of EMin only considers costs due to events within a certain distance defined by a kernel of size κ^d, thus reducing the complexity from N_e^2 to $N_e\kappa^d$, where $\kappa \ll N_e$.

Method	Time complexity
CMax [7]	$\mathcal{O}(N_e)$
AEMin [21]	$\mathcal{O}(N_e\kappa^d)$
EMin [21]	$\mathcal{O}(N_e^2)$
Poisson Point-Process (ours)	$\mathcal{O}(N_e)$

Table 3. **Time complexity** of each algorithm as a function of the number of input events N_e and kernel size κ^d.

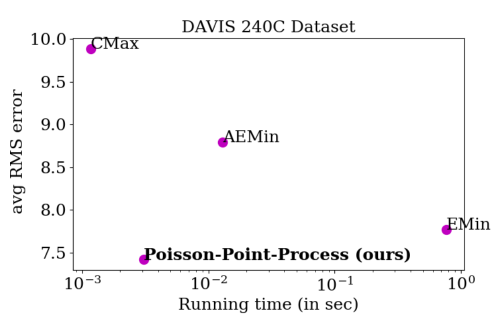


Figure 4. **Runtime vs. accuracy comparison.** Time for one loss function evaluation given a fixed set of 30k events versus accuracy measured in terms of average RMS-error across boxes, poster and dynamic. Time plotted on log-scale.

4.3. Ablation study

Poisson rate parameter λ - the expected rate of events. In Section 3.4 we have described our optimization problem as a maximum likelihood procedure: the likelihood of aligned data modeled as a Poisson point process $\text{Pois}(\lambda_{\mathbf{x}})$ is higher than the likelihood of unaligned data under our model. Computing the likelihood of events requires knowledge of the rate parameter λ. Here, we discuss two options

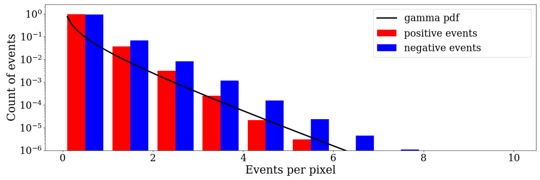


Figure 5. **Prior distribution over λ.** Histogram of expected event counts per pixel $\lambda_{\mathbf{x}}$ during a time interval ΔT (events with positive polarity in red and negative in blue). PDF of the Gamma distribution shown overlaid.

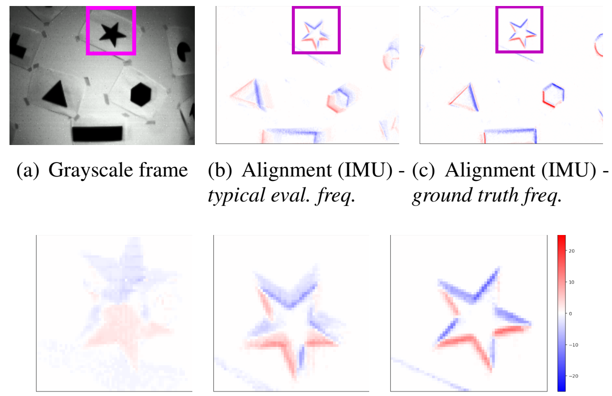
to deal with this unknown parameter: (i) marginalizing it out, (ii) using its per-pixel ML-estimate.

Marginalizing out λ. Integrating over λ leads to a negative binomial distribution (Eq. 8) - which is also often described as a gamma-Poisson mixture with a gamma distribution as the mixing distribution. We derive our prior distribution $\lambda \sim \text{Gamma}(r, q^{-1}(1-q))$ from observed (unaligned) event data. In particular, the data is the expected counts of events per pixel during a time interval ΔT. Both parameters r and q defining the Gamma distribution are obtained via maximum likelihood estimation. Fig. 5 shows the histogram of expected event counts per pixel for a set of 30000 events. The best fitting Gamma distribution with parameters $r = 0.1$ and $q = 0.39$ is shown overlaid.

Per pixel ML estimate, $\lambda_{\mathbf{x}}$. Given a set of events at a particular pixel location $\mathbf{x}$, the ML estimate for the rate parameter $\lambda_{\mathbf{x}}$ is simply the *count* of events at that location (since we just have one observation sample). This approach might have the advantage of capturing the scene structure, where a point in the world triggers events at different rates. However due to the relatively small sample size this approach is less robust than integrating over the unknown variable.

Overall both approaches perform well, but marginalizing the unknown variable out seems to be more robust on average. Using an ML estimate for λ leads to an average RMS error of 12.1 deg/s, marginalisation improves slightly and reaches an RMS error of 12.0 deg/s.

Affine model of angular velocity. Event alignment of a fixed batch of events (e.g., 30k) is typically done via assuming a constant velocity during the time span of the events. However, such a time span is a variable that depends on the amount of texture in the scene. As the last row of Fig. 3 shows, 30k events is too many for low-textured scenes (shapes). A possible fix to this issue is to use an adaptive number of events, depending on texture [16]. However this makes comparisons more difficult to interpret. We develop a different solution: using a more expressive motion model. Fig. 6 shows that for a large interval ΔT a high quality alignment can only be achieved with more complex (but smooth) velocity estimates, such as the ground truth signal $\omega(t)$. Since alignment with constant velocity $\omega(t) \approx \omega_0 \; \forall t \in [0, \Delta T]$ is not enough, we propose a higher order model $\omega(t) \approx \omega_0 + at$ (affine), thus estimating $(\omega_0, a) \in \mathbb{R}^6$. This improves event alignment (Fig. 6(f)). A typically evaluation strategy is to compare the estimated *constant* velocity with its closest ground truth. This allows for evaluating average angular velocity over a fixed time interval, but leads to inaccurate event alignment as can be seen in Figure 6(b). To mitigate this issue evaluating angular velocity using the high frequency (1kHz) of the ground truth provided by [20] is required.



(a) Grayscale frame (b) Alignment (IMU) - *typical eval. freq.* (c) Alignment (IMU) - *ground truth freq.*

(d) Unaligned (e) Alignment (our) - *const. velocity* (f) Alignment (our) - *affine velocity model*

Figure 6. **Affine velocity model**. Quality of event alignment for constant velocity estimates within ΔT compared to *smooth* velocity model estimates from our affine velocity model.

5. Conclusion

Inspired by congealing [13], this paper has introduced a new probabilistic approach for event alignment. In particular we model the aligned events as independent, *per pixel Poisson point processes*, or a spatio-temporal Poisson point process. Based on this idea, we derive a likelihood function for a set of observed (unaligned) events and maximize it to estimate the camera motion that best explains the events. This method leads to new state-of-the-art results for angular velocity estimation, with only 0.5% relative RMS error with respect to the velocity excursion. Our event alignment method is not specific of rotational motion, as we have demonstrated how it can be applied to other types of motion (e.g., translational). This opens the door to utilize our method for solving related event alignment problems, such as motion segmentation [30] and feature tracking [29], which in turn enable higher level scene understanding.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.

References

[1] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240x180 130dB 3μs latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits*, 49(10):2333–2341, 2014. 2, 5

[2] Matthew Cook, Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika Steger. Interacting maps for fast visual interpretation. In *Int. Joint Conf. Neural Netw. (IJCNN)*, pages 770–776, 2011. 2

[3] Tobi Delbruck, Vicente Villanueva, and Luca Longinotti. Integration of dynamic vision sensor with inertial measurement unit for electronically stabilized event-based vision. In *IEEE Int. Symp. Circuits Syst. (ISCAS)*, pages 2636–2639, 2014. 2, 6

[4] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, Hirotsugu Takahashi, Hayato Wakabayashi, Yusuke Oike, and Christoph Posch. A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86μm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline. In *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2020. 1

[5] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 1

[6] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 12272–12281, 2019. 2, 3, 6

[7] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3867–3876, 2018. 2, 3, 5, 6, 7

[8] Guillermo Gallego and Davide Scaramuzza. Accurate angular velocity estimation with an event camera. *IEEE Robot. Autom. Lett.*, 2(2):632–639, 2017. 2, 5

[9] Gary B. Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *Int. Conf. Comput. Vis. (ICCV)*, 2007. 3

[10] Gary B. Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems 25*, pages 773–781, 2012. 3

[11] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J. Davison. Simultaneous mosaicing and tracking with an event camera. In *British Mach. Vis. Conf. (BMVC)*, 2014. 2

[12] Xavier Lagorce, Garrick Orchard, Francesco Gallupi, Bertram E. Shi, and Ryad Benosman. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7):1346–1359, July 2017. 1

[13] Erik Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(2):236–250, 2006. 2, 3, 8

[14] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits*, 43(2):566–576, 2008. 1, 2

[15] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally optimal contrast maximisation for event-based motion estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020. 2, 3, 5

[16] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Mach. Vis. Conf. (BMVC)*, 2018. 8

[17] Marwan Mattar, Michael Ross, and Erik Learned-Miller. Nonparametric curve alignment. In *Int. Conf. Acoust., Speech, Signal Proc. (ICASSP)*, 2009. 3

[18] Erik G. Miller. *Learning from one example in machine vision by sharing probability densities*. PhD thesis, Massachusetts Institute of Technology, 2002. 3

[19] Anton Mitrokhin, Cornelia Fermuller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018. 2, 3

[20] 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. *Int. J. Robot. Research*, 36(2):142–149, 2017. 1, 5, 6, 8

[21] Urbano Miguel Nunes and Yiannis Demiris. Entropy minimisation framework for event-based vision model estimation. In *Eur. Conf. Comput. Vis. (ECCV)*, 2020. 2, 3, 5, 6, 7

[22] Chethan M Parameshwara, Nitin J Sanket, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. 0-mms: Zero-shot multi-motion segmentation with a monocular event camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 3

[23] Xin Peng, Ling Gao, Yifu Wang, and Laurent Kneip. Globally-optimal contrast maximisation for event cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–1, 2021. 3

[24] Xin Peng, Yifu Wang, Ling Gao, and Laurent Kneip. Globally-optimal event camera motion estimation. In *Eur. Conf. Comput. Vis. (ECCV)*, 2020. 3

[25] 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 J. Solid-State Circuits*, 46(1):259–275, Jan. 2011. 1

[26] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. 1

[27] Christian Reinbacher, Gottfried Munda, and Thomas Pock. Real-time panoramic tracking for event cameras. In *IEEE Int. Conf. Comput. Photography (ICCP)*, pages 1–9, 2017. 2, 5

[28] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios. *IEEE Robot. Autom. Lett.*, 3(2):994–1001, Apr. 2018. 5

[29] Hochang Seok and Jongwoo Lim. Robust feature tracking in DVS event stream using Bezier mapping. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2020. 8

[30] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *Int. Conf. Comput. Vis. (ICCV)*, pages 7243–7252, 2019. 2, 3, 8

[31] Timo Stoffregen and Lindsay Kleeman. Event cameras, contrast maximization and reward functions: An analysis. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 3

[32] Roy L. Streit. *Poisson Point Processes*. Springer, 2010. 4, 5

[33] Yunjae Suh, Seungnam Choi, Masamichi Ito, Jeongseok Kim, Youngho Lee, Jongseok Seo, Heejae Jung, Dong-Hee Yeo, Seol Namgung, Jongwoo Bong, Jun seok Kim, Paul K. J. Park, Joonseok Kim, Hyunsurk Ryu, and Yongin Park. A 1280x960 Dynamic Vision Sensor with a 4.95-μm pixel pitch and motion artifact minimization. In *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2020. 1

[34] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 5816–5824, 2017. 5

[35] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems (RSS)*, 2018. 1

[36] Lilla Zollei, Erik Learned-Miller, W. Eric. L. Grimson, and William Wells. Efficient population registration of 3D data. In *Workshop on Computer Vision for Biomedical Image Applications: Curre nt Techniques and Future Trends*, 2005. 3