Ego Motion Compensation in Event Camera Using Forward Prediction

Sumit Mahesh Khobragade

A thesis submitted for the degree of

Master of Engineering

The Australian National University

College of Engineering, Computing and Cybernetics

September 2025

To family, friends, and the wonderful universe we are intimately connected with

Disclaimer

I hereby declare that the work undertaken in this thesis has been conducted by me alone, except where indicated in the text. I conducted this work between October 2019 and March 2023, during which period I was a PhD student at the Australian National University. This thesis, in whole or any part of it, has not been submitted to this or any other university for a degree.

This thesis has been compiled as a Thesis by Compilation in accordance with relevant ANU policies. Each of the 2 main chapters in this thesis have been published in peer reviewed journals. I have made significant contribution to each of these journal articles and have written the text of the papers myself, except where indicated otherwise.

Signature

Sumit Mahesh Khobragade

30 October 2025

Acknowledgements

Abstract

Event cameras sense per-pixel brightness changes asynchronously with microsecond latency and very high dynamic range, offering a promising alternative to conventional frame-based vision in robotics. However, during ego-motion even static scenes generate dense background events, mixing with true object-motion signals and creating bandwidth and processing overhead. Suppressing these predictable ego-motion events at the source could significantly reduce redundancy and improve efficiency in event-based perception.

This thesis explores a forward-prediction approach to cancel ego-motion events caused by rotational motion. Each event is propagated to a predicted future location using an estimated rotation centre and angular velocity, where an inverse-polarity event is emitted to suppress redundant observations. The system is designed to operate in real time without relying on batch accumulation or image reconstruction. Using a controlled spinning-disc dataset, cancellation performance was evaluated under varying parameters including prediction horizon, spatial and temporal tolerances, polarity handling, and model bias. Results demonstrate cancellation rates up to 84% at short horizons, with sensitivity strongly linked to parameter choice and estimation accuracy.

While current performance highlights limitations of simplified motion models and sensor noise characteristics, the study establishes a structured methodology for analysing residual event distributions and parameter effects. This provides a foundation for future extensions to general ego-motion, uncertainty-aware prediction, and hardware-efficient implementations aimed at enabling practical event-level suppression in robotics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview and Motivation

Event cameras (also called dynamic vision sensors) are bio-inspired visual sensors that report per-pixel brightness changes asynchronously with microsecond latency and very high dynamic range, rather than periodic intensity frames Gallego et al. (2020). Compared with conventional frame cameras, their properties—high temporal resolution, reduced motion blur, high intra-scene dynamic range, and low power—make them particularly attractive for high-speed robotics and perception in challenging illumination Gallego et al. (2020); Lichtsteiner et al. (2008); Brandli et al. (2014). The first commercially available device, the DVS, demonstrated $\sim$15 µs latency and $\sim$120 dB dynamic range Lichtsteiner et al. (2008); later DAVIS sensors combined asynchronous events with global-shutter frames in the same pixel array, further broadening applications Brandli et al. (2014).

A fundamental consequence of the event sensing principle is that camera *ego-motion* over a static scene moves edges across the image plane and triggers dense background events ("ego events"). These ego events can dominate the event stream and confound downstream algorithms (e.g., tracking, segmentation, or mapping) that focus on independently moving objects Stoffregen et al. (2019); Gallego et al. (2020). In practice, many pipelines therefore first estimate the camera motion and then compensate or warp events to reduce blur in the accumulated Image of Warped Events (IWE), improving the signal-to-noise structure of event data Gallego et al. (2018); Bardow et al. (2016); however, higher event rates also increase bandwidth use and processing load for transmitting and handling events.

This thesis explores an alternative approach: *proactive* suppression of ego-motion events via **forward prediction and per-event cancellation**. Instead of accumulating events and compensating them in batches, we model rotational ego-motion, predict where an event will reappear after a short horizon Δt, and *software-level cancellation* (dropping matched pairs from the event stream) at that predicted location when the horizon elapses, thereby suppressing predictable, redundant observations *as they arrive.* Such anticipation echoes predictive-coding principles in biological vision, where neural circuits suppress expected input to highlight novel signals Hosoya et al. (2005); Rao & Ballard (1999).

1.2 Problem Statement and Research Questions

Consider a scene observed by an event camera undergoing planar circular motion about a fixed rotation centre. Even if the scene is static, the sensor produces a high rate of ego-motion events due to apparent edge motion. Let Δt be a short prediction horizon. The core question is: *can ego-motion–induced events be predicted and cancelled causally to reduce bandwidth and computational load without affecting unpredictable (object-motion) events?* Specifically, we ask:

1. How sensitive is cancellation to errors in the assumed rotation centre and angular velocity used by the predictor? Gallego & Scaramuzza (2017); Gallego et al. (2018)

2. What cancellation performance (cancellation ratio, residual density relative to background) is achievable as a function of algorithm parameters (e.g., Δt, spatial/temporal tolerances) and sensor effects (polarity asymmetry, contrast threshold variability)? Gallego et al. (2020); Xu et al. (2020)

3. How does proactive cancellation compare qualitatively to standard motion compensation (e.g., contrast-maximization/IWE) on the same sequences? Gallego et al. (2018); Bardow et al. (2016); Stoffregen et al. (2019)

1.3 Background: Event Cameras and Ego-Motion Compensation

Event generation and noise. An event (x, y, t, p) is raised when the change in log-intensity at pixel (x, y) exceeds a positive or negative contrast threshold, with polarity $p \in \{+1, -1\}$ Gallego et al. (2020). Practical devices exhibit per-pixel threshold variations and background activity that act as a noise floor, and finite photoreceptor response can introduce small timing offsets; these factors influence any matching/cancellation tolerance Gallego et al. (2020). The DVS architecture and its DAVIS follow-ups provide concrete hardware context and limits Lichtsteiner et al. (2008); Brandli et al. (2014).

Motion estimation with events. A large body of work estimates ego-motion or optical flow directly from events. Variational methods simultaneously recover flow and latent intensity Bardow et al. (2016). The *contrast maximization* framework formalizes motion estimation as maximizing the sharpness of an IWE after warping events with motion parameters, and has been shown to unify solutions for motion, depth, and flow Gallego et al. (2018). For rotational motion in particular, accurate angular velocity estimation from events has been demonstrated Gallego & Scaramuzza (2017). In parallel, learning-based methods can infer optical flow, depth, and egomotion from events by enforcing motion-compensation losses Zhu et al. (2019). More recently, global or robust formulations improve motion compensation and reduce artifacts Xu et al. (2020).

Why cancellation instead of compensation? Most pipelines accumulate events over a window and then compensate them (post hoc) to form sharp IWEs for subsequent tasks Gallego et al. (2018); Stoffregen et al. (2019). In bandwidth- or latency-critical systems, however, upstream *suppression* of predictable ego-events could save downstream compute and highlight residual, task-relevant signals (e.g., truly independent movers). The proposed forward-prediction approach adopts short, causal horizons and emits anti-events, drawing conceptual inspiration from inhibitory circuits and predictive coding in the retina and cortex Hosoya et al. (2005); Rao & Ballard (1999) while remaining purely algorithmic and sensor-level.

1.4 Methodological Scope and Assumptions

Scope and rationale. To make the problem tractable, measurable, and reproducible, we study ego-motion cancellation in a controlled **circular-motion** setting using a spinning-disc apparatus. Rotational setups are common in event-camera evaluation and provide analytic motion fields that simplify modeling, identification, and ground-truth checks Gallego & Scaramuzza (2017); Stoffregen et al. (2019). The goal is to test whether ego-motion–induced events can be *proactively* suppressed via forward prediction and per-event cancellation, without relying on batch accumulation or dense image-of-warped-events (IWE) optimization Gallego et al. (2018); Bardow et al. (2016).

Motion model. We assume an approximately static scene and predominant rotational ego-motion of the sensor relative to the disc. Let (c_x, c_y) denote the rotation center on the image plane and ω the signed angular velocity. For an event $e = (x, y, t, p)$ we predict its future location after a short horizon Δt by rotating about (c_x, c_y) using the estimated ω; full details and derivations are deferred to Chapter 4. The horizon Δt is kept short (sub-millisecond to a few milliseconds) to limit phase error and minimize model drift Gallego & Scaramuzza (2017); Gallego et al. (2018).

Per-event cancellation mechanism. At physical time $t + \Delta t$, we perform a spatio-temporal match around $(x', y', t+\Delta t)$ within tolerances $(\epsilon_{xy}, \epsilon_t)$ and with *opposite polarity* to the original event. Upon a successful match, we remove both the predicted and matched events from the stream (software-level cancellation), thereby suppressing the predicted observation. This preserves the asynchronous, low-latency nature of event processing and avoids large buffers or global optimization windows Bardow et al. (2016); Gallego et al. (2018).

Metrics and evaluation. We report (i) **cancellation ratio** (fraction of events cancelled), (ii) **residual event density** (events/area) on disc vs. background regions, (iii) **radial residual profiles** to see where cancellation fails along the disc radius, and (iv) **ablation curves** vs. parameter settings. These metrics echo common practice in motion compensation and event denoising/warping evaluations Bardow et al. (2016); Gallego et al. (2018); Xu et al. (2020). Where relevant, we interpret residual changes as effective SNR improvements against a background-activity baseline Gallego et al. (2020).

Sensitivity study (design). To understand robustness, we systematically vary:

- **Prediction horizon** Δt: e.g., 0.25, 0.5, 1, 2, 4 ms (phase-mismatch grows with Δt).
- **Spatial tolerance** ϵ_{xy}: e.g., 0.5–3 pixels (trade-off between true matches and false positives).
- **Temporal tolerance** ϵ_t: e.g., $50 - 500\,\mu$s (timestamp jitter vs. strictness).
- **Polarity handling**: strict opposite-polarity vs. relaxed polarity checks (to probe asymmetries).
- **Motion-estimation bias**: add small biases to ω and/or (c_x, c_y) to quantify sensitivity.

We report cancellation curves and residual maps for each sweep and analyze trends (Chapter 8).

Assumptions and limitations.

1. **Scene stationarity**: the background is static; residual non-stationarity is treated as object/scene motion.
2. **Short-horizon linearizability**: Δt is small enough that unmodeled effects (e.g., small translation, acceleration, rolling-shutter timing) are negligible to first order Gallego & Scaramuzza (2017).
3. **Calibration fidelity**: intrinsic parameters and distortion are either negligible at the ROI or pre-compensated; the disc center is near the true rotational center in the image.
4. **Sensor non-idealities**: background activity, threshold mismatch, and refractory effects exist Brandli et al. (2014); Delbrück et al. (2020); we treat them as noise and characterize them via a background baseline.
5. **Controlled lighting**: flicker is minimized; where present, it is acknowledged as a confound and included in residual analysis Gallego et al. (2020).

These assumptions bound the scope of claims and clarify failure modes (e.g., longer Δt will degrade cancellation due to phase error; miscentered (c_x, c_y) yields radius-dependent residuals).

1.5 Contributions

This thesis does not claim external publications; the contributions are *methodological, analytical, and empirical*:

1. **A causal, per-event forward-prediction cancellation pipeline for rotational ego-motion.** A lightweight software method that predicts future event locations using an estimated (c_x, c_y, ω), then suppresses predictable ego-motion events by matching and removing predicted–real event pairs at horizon Δt, while preserving per-event latency (no dense IWEs or batch optimization) Bardow et al. (2016); Gallego et al. (2018).

2. **A principled sensitivity analysis of key parameters.** Systematic sweeps over Δt, spatial/temporal tolerances, polarity handling, and motion-estimation biases, with quantitative effects on cancellation ratio and residual structure; results are interpreted in terms of phase mismatch, timing uncertainty, and model error (Chapter 8).

3. **Evaluation metrics and residual analysis.** Clear definitions and measurements of cancellation ratio, residual event density, radial residual profiles, and comparisons to a background-activity baseline, enabling SNR-like interpretation and fair comparisons to motion compensation baselines Gallego et al. (2020); Bardow et al. (2016); Gallego et al. (2018); Xu et al. (2020).

4. **Controlled spinning-disc testbed and empirical study.** A reproducible evaluation on rotational sequences (and sequences with moderate departures from ideal circular motion), including qualitative overlays and quantitative ablations versus standard motion-compensation procedures Gallego et al. (2018); Bardow et al. (2016); Stoffregen et al. (2019).

5. **Open implementation details.** Implementation choices (data structures, buffering, complexity, and parameter defaults) documented to support reproduction and future extension to general ego-motion.

Together, these contributions position proactive cancellation as an alternative approach to existing alignment/estimation methods in event vision Bardow et al. (2016); Rebecq et al. (2017); Gallego et al. (2018) and motivate future work on uncertainty-aware prediction and hardware-level implementations (e.g., FPGA/MCU for on-sensor processing).

1.6 Thesis Outline

This thesis follows the structure established in event-vision literature and our research group's practice:

Chapter 2 — Background & Related Work. Event-camera principles (event generation model, polarity, contrast threshold), sensor non-idealities (background activity, threshold mismatch, flicker), processing paradigms (asynchronous vs. batch), and applications in flow, VO/SLAM, and motion compensation. We summarize contrast maximization/IWE methods and prior work most related to ego-motion handling Gallego et al. (2020); Bardow et al. (2016); Gallego et al. (2018).

Chapter 4 — Problem Formulation. Notation, circular-motion geometry, forward-prediction model, cancellation objective, and evaluation metrics (cancellation ratio, residual densities, radial profiles, background baseline). We formalize matching in (x, y, t) with polarity constraints and tolerances.

Chapter 3 — Motion Estimation & Prediction. Estimation of (c_x, c_y, ω) from events (and optional cues), prediction of future event locations, and practicalities (timestamping, distortion, ROIs, buffering). We discuss sources of bias/variance and their expected effects on prediction accuracy Gallego & Scaramuzza (2017).

Chapter 5 — Event Cancellation Algorithm. Full pipeline description: data structures (grid/k-d tree), matching policy, polarity handling, optional pre-blur of predicted events, complexity, and runtime considerations. We contrast this with batch motion-compensation against a dense objective Gallego et al. (2018).

Chapter 8 — Sensitivity Analysis of Parameters. Experimental design and results for parameter sweeps: Δt, ϵ_{xy}, ϵ_t, polarity, and ω bias. We provide cancellation curves, residual maps, and robustness analyses, and relate observed trends to modeling assumptions.

Chapter 6 — Experimental Setup. Hardware (event sensor, disc rig, lighting), calibration notes, datasets and protocols, and software pipeline (preprocessing, estimation, prediction, cancellation, metrics) with implementation details for reproducibility.

Chapter 7 — Results. Quantitative and qualitative evaluations: overall cancellation ratios, residual densities, radial profiles, ablations vs. motion-compensation baselines, and representative overlays/time-slices; discussion of failure cases (e.g., longer horizons, miscentered rotation).

Chapter 9 — Discussion. Interpretation of sensitivity results; implications for event-based perception; limitations (model mismatch, noise, timing offsets) and threats to validity.

Chapter 10 — Conclusions & Future Work. Summary of findings and contributions; concrete directions toward general ego-motion, uncertainty-aware forward prediction, polarity/asymmetry modeling, and potential hardware-level implementations for on-sensor processing.

Chapter 2

Background and Related Work

2.1 Principles of Event Cameras

Event cameras, also known as neuromorphic vision sensors, represent a departure from the conventional frame-based paradigm by asynchronously reporting per-pixel brightness changes. The first commercially available Dynamic Vision Sensor (DVS) was introduced by Lichtsteiner et al. Lichtsteiner et al. (2008), and has since inspired a series of improved designs including DAVIS (Dynamic and Active-pixel Vision Sensor) Brandli et al. (2014) and ATIS (Asynchronous Time-based Image Sensor) Posch et al. (2014). In contrast to frame cameras, which operate at fixed frame rates and expose all pixels simultaneously, each pixel in an event camera operates independently and emits an event whenever the change in log-intensity exceeds a preset threshold. Each event is encoded as a tuple $e = (x, y, t, p)$, consisting of pixel coordinates (x, y), timestamp t, and polarity $p \in \{+1, -1\}$ indicating whether the brightness increased or decreased.

This architecture offers several key advantages for robotics and perception tasks. First, the temporal resolution of event cameras is on the order of microseconds, several orders of magnitude faster than conventional cameras that typically operate at 30–60 frames per second. This enables accurate capture of extremely fast motion without motion blur. Second, their dynamic range exceeds 120 dB, allowing operation in high-contrast environments such as driving from a dark tunnel into bright sunlight, where standard cameras would saturate or underexpose Gallego et al. (2020). Third, event cameras have significantly lower power consumption, since pixels remain idle until a brightness change occurs, making them attractive for embedded and mobile robotics applications.

Despite these benefits, event cameras also exhibit non-idealities that must be considered when designing algorithms. Real devices produce spurious events due to sensor noise, thermal drift, and leakage currents, commonly referred to as background activity Gallego et al. (2020). Per-pixel contrast thresholds vary and drift over time, introducing mismatch in sensitivity across the sensor array. Polarity asymmetries mean that positive and negative events may not be triggered symmetrically for equal magnitude changes. Additionally, refractory effects impose a minimum dead time after an event before a pixel can trigger again Delbrück et al. (2020). These imperfections result in a noisy stream that complicates downstream processing and can degrade performance if left unaddressed. Moreover, because events encode temporal contrast rather than intensity frames, they are not immediately compatible with many conventional computer-vision pipelines; there is still no universally accepted representation or processing recipe for event data Gallego et al. (2020). For example, Wang (2025) notes that background activity events can dominate low-texture regions, producing clutter that needs to be filtered or suppressed before higher-level processing Wang (2025). Scheerlinck Scheerlinck (2021) similarly highlights that sensor noise is a major limiting factor in low-light event-based reconstruction tasks.

Another fundamental characteristic of event cameras is their data-dependent bandwidth. Since events are generated only where brightness changes occur, the event rate scales with both scene dynamics and motion speed. Static scenes under constant illumination produce very few events, whereas highly dynamic scenes can generate millions of events per second. Recent high-resolution event sensors, such as Prophesee's Gen4 and Sony's DAVIS346, can output hundreds of megabytes per second under aggressive motion conditions Finateu et al. (2020). This property is both a strength and a challenge: it enables efficient encoding of salient changes but requires careful bandwidth management in systems that must operate under real-time constraints.

In summary, event cameras provide a fundamentally new data modality characterized by high temporal resolution, wide dynamic range, and sparse, asynchronous output. These properties have made them increasingly popular in applications requiring high-speed and robust visual sensing. However, their non-idealities and data-dependent bandwidth necessitate the development of specialized algorithms to exploit their advantages while mitigating their limitations.

2.2 Event Data Processing: Asynchronous vs Batch Paradigms

Given the unique nature of event data, two main paradigms have emerged for processing: (i) asynchronous, per-event methods and (ii) batch-based approaches that accumulate events into spatio-temporal representations. Beyond motion compensation, both paradigms underpin a variety of applications spanning classical vision pipelines, learning-based models, and hybrid filtering Gallego et al. (2020).

2.2.1 Asynchronous, per-event methods

Asynchronous pipelines process events as they arrive, preserving the microsecond-level latency advantage of event sensors. Methods include classical filtering and tracking (e.g., Kalman-like updates), per-event mapping/control, and learning-based models that directly ingest streams (spiking or recurrent nets) Maqueda et al. (2018); Gallego et al. (2020). Such pipelines are attractive when reaction time is critical because they avoid window accumulation.

However, asynchronous methods can struggle to incorporate global spatial context because they operate on local neighborhoods rather than integrated windows. This can lead to susceptibility to noise and difficulties in handling large-scale structures or low-texture regions where events are sparse. Prior work notes that purely asynchronous pipelines must balance responsiveness against stability, often requiring additional filtering or regularization to avoid drift Scheerlinck (2021); Gallego et al. (2020).

2.2.2 Batch accumulation and windowed representations

Batch-based approaches accumulate events over short temporal windows and transform them into image- or volume-like representations. Common choices include event-count images (histograms), time surfaces/surfaces of active events, and voxel grids for learning-based models Gallego et al. (2020). For motion estimation, a task-specific representation is the Image of Warped Events (IWE), where events are spatially warped according to candidate motion parameters and accumulated; correct parameters yield sharp structures and incorrect ones produce blur Gallego et al. (2018).[1] This view enables contrast maximization: optimizing motion parameters to maximize IWE sharpness. Bardow et al. demonstrated joint flow and intensity recovery Bardow et al. (2016).[2] Subsequent work framed optical flow, depth, and rotation within this objective Gallego et al. (2018). Multi-motion segmentation can also be posed in this setting by comparing sharpness under competing motion models Stoffregen et al. (2019). In parallel, learning-based pipelines use windowed event tensors (e.g., voxel grids) for recognition or reconstruction Rebecq et al. (2019); Gallego et al. (2020).

Batch methods leverage spatial context and can achieve high-quality reconstructions, making them suitable for tasks such as VO/VIO/SLAM and recognition. Rebecq et al. integrated accumulated-event objectives with inertial data to achieve accurate VIO pipelines Rebecq et al. (2017, 2019). These results indicate that windowed processing can rival frame-based methods under high-speed or high dynamic range conditions Gallego et al. (2020).

2.2.3 Positioning of this thesis

The choice between asynchronous and batch paradigms depends on task requirements and latency constraints. Broadly, windowed methods provide stronger global context at

[1] Gallego et al. describe “accumulate them into frames” and “warp them to a reference time” to form an image of warped events (IWE).

[2] Bardow et al. formulate a “sliding window” variational model and report “an accumulation of all events during a period of 33 ms,” illustrating windowed processing.

the expense of additional delay (e.g., contrast-maximization and IWE-based objectives optimize over short event packets) Gallego et al. (2018); Bardow et al. (2016); Stoffregen et al. (2019); Rebecq et al. (2017, 2019), while per-event methods prioritize responsiveness and microsecond timing Gallego et al. (2020). This thesis positions itself within the asynchronous paradigm, focusing on per-event forward prediction and cancellation. Unlike batch methods that warp events into alignment, our approach proactively predicts and suppresses ego-motion events as they arrive, aiming to preserve the low-latency advantage of event cameras while reducing clutter from predictable motion.

2.3 Motion Estimation and Ego-Motion

Estimating the motion of a moving camera (ego-motion) from event streams is a foundational capability for navigation and scene understanding. The problem is challenging because the observed event patterns conflate the effects of camera motion with those of independent object motion and illumination changes. Nevertheless, a large body of work has shown that event cameras can yield accurate motion estimates even in regimes that are difficult for standard frame-based sensors Gallego et al. (2020). Approaches fall broadly into four families: (i) local optical-flow estimation, (ii) global motion compensation via contrast maximization and images of warped events (IWEs), (iii) state-estimation pipelines for visual odometry and SLAM (often fused with inertial sensing), and (iv) learning-based models that operate on event frames/voxels or time surfaces to infer flow, depth, or reconstructions Bardow et al. (2016); Gallego et al. (2018); Rebecq et al. (2017, 2019); Zhu et al. (2019); Gallego et al. (2020).

2.3.1 Local optical flow

Local methods estimate the instantaneous image velocity at each point using spatio-temporal neighborhoods of events. Plane-fitting in the x–y–t event space is a classic approach, interpreting events lying on a tilted plane as evidence of a consistent motion direction and speed Benosman et al. (2014). Variants adopt gradient constraints, spatio-temporal filters, or probabilistic models to improve robustness to noise and sparse data Gallego et al. (2020). While local flow is informative for dynamic textures and tracking, it can be noisy in low-texture regions and does not yield a single coherent ego-motion estimate in rigid scenes. Moreover, local estimates are susceptible to the *aperture problem*: when only a small edge fragment is visible through a local "aperture," motion parallel to the edge is ambiguous and only the normal (perpendicular) component can be recovered; this necessitates priors/regularization or larger spatial support to obtain globally consistent fields Gallego et al. (2020).

2.3.2 Global motion compensation and contrast maximization

Global approaches exploit the fact that if a set of motion parameters is correct, warping events into a common reference yields sharp, high-contrast structures; incorrect parameters smear events and reduce contrast. This insight underpins the IWE and contrast-maximization family of methods Gallego et al. (2018). Bardow et al. proposed SOFIE, which jointly estimated optical flow and intensity by maximizing alignment-based objectives over short event windows Bardow et al. (2016). Gallego et al. formalized contrast maximization as a unifying framework for motion, depth, and rotation estimation, showing that diverse tasks can be framed as maximizing a sharpness/energy functional of warped events Gallego et al. (2018). Xu et al. improved robustness with smoothness constraints and regularization, further stabilizing compensation in challenging conditions Xu et al. (2020). These methods often provide high-quality, globally consistent estimates but rely on accumulating events into windows, introducing latency that partially offsets the intrinsic temporal advantages of event sensing.

2.3.3 Event-based VO, VIO, and SLAM

A third line of work integrates motion models into state-estimation pipelines, combining events with inertial measurements and (in some cases) frames. Rebecq et al. demonstrated event-based visual odometry with photometric consistency and later extended to VIO by fusing IMU data, achieving accurate trajectories in fast and high dynamic range scenarios Rebecq et al. (2017, 2019). Subsequent pipelines have incorporated bundle-adjustment-like optimization, spatio-temporal feature tracking, and probabilistic filtering to enhance robustness Gallego et al. (2020). These results establish that event cameras can deliver reliable ego-motion estimation in regimes where conventional cameras would suffer from blur or saturation. However, they still treat ego-motion *post hoc*—estimating it and then compensating the stream—rather than suppressing predictable events as they arrive.

2.3.4 Implications for cancellation

Across these families, a common theme emerges: ego-motion produces dense, structured event patterns as scene edges sweep the sensor; motion compensation sharpens these structures and thereby reveals independently moving objects Stoffregen et al. (2019). In this thesis, we adopt an alternative stance: instead of aligning events to measure motion, we use motion to *predict* and proactively *cancel* ego-motion events at the per-event timescale. The rationale is to reduce bandwidth and clutter early—before downstream modules (flow, VO/SLAM, or recognition) have to process redundant events—while preserving the low latency of asynchronous pipelines Gallego et al. (2018); Bardow et al. (2016).

2.4 Rotational and Circular-Motion Setups

Controlled rotational rigs (e.g., spinning discs, wheels, turntables) are widely used for evaluation and calibration in event-based vision because they induce large, well-structured patterns with simple analytic motion fields Gallego & Scaramuzza (2017); Stoffregen et al. (2019). Under pure rotation about a center (c_x, c_y) on the image plane and angular velocity ω, the normal flow at a pixel depends linearly on the distance to the center, and the trajectories of edge points follow circular arcs. This analytic structure makes rotation an ideal stress test for algorithms that must handle high apparent velocities and tightly curved trajectories Gallego & Scaramuzza (2017).

2.4.1 Why rotation is a good testbed

First, rotational motion excites strong edge responses across the full field of view, generating dense event patterns that probe the limits of event bandwidth and algorithm throughput. Second, the geometry provides a compact parametric description—rotation center and angular velocity—that can be estimated and then used for forward prediction. Third, circular motion is repeatable and easy to instrument: spinning-disc apparatuses can run at constant speed for long durations, enabling statistically meaningful evaluation across parameter sweeps Stoffregen et al. (2019). In practice, many event-vision papers use rotations explicitly to demonstrate robustness to high-speed motion and to validate compensation/flow methods Gallego et al. (2018).

2.4.2 Relation to motion compensation and segmentation

Stoffregen et al. showed that motion compensation, when applied to scenes with both camera and object motion, can be used to segment independently moving objects: a correct background motion model aligns background events (increasing IWE contrast) but fails to align foreground, which remains diffuse Stoffregen et al. (2019). This is conceptually close to the motivation for ego-motion cancellation: by explaining away background motion, one increases the salience of object-driven events. The key difference is operational timing: segmentation-by-compensation aligns events after accumulation, while our cancellation aims to reduce background events at the per-event timescale.

2.4.3 Operational use in this thesis

We use a spinning-disc setup as a controlled environment in which circular motion is an appropriate model. This enables us to (i) estimate (c_x, c_y, ω) from the event stream or auxiliary instrumentation, (ii) forward-predict event locations by a short horizon Δt, and (iii) evaluate cancellation performance as a function of Δt, spatial/temporal tolerances, polarity checks, and motion-estimation biases. Rotational setups thereby serve both as a validation platform and as a diagnostic lens to study sensitivity to modeling errors and timing jitter Gallego & Scaramuzza (2017); Gallego et al. (2018).

2.5 Event Prediction, Forward Models, and Predictive Suppression

Forward prediction is a natural outgrowth of motion estimation: if a model explains how events move, it can be used not only to *align* events retrospectively but also to *forecast* where they will occur. In the short-horizon regime relevant to event cameras, such forecasts can be highly accurate if the motion model and parameters are adequate. This observation has surfaced implicitly in contrast-maximization pipelines (e.g., evaluating candidate motions over windows) and explicitly in learning-based event extrapolation methods Gallego et al. (2018, 2020). In our context, we leverage forward prediction to generate inverse-polarity "anti-events" intended to cancel predictable ego-motion events before they inflate bandwidth and obscure object motion.

2.5.1 Short-horizon propagation

For small horizons Δt (sub-millisecond to a few milliseconds), a circular-motion model parameterized by (c_x, c_y, ω) can propagate an event $e = (x, y, t, p)$ to a predicted location (x', y') at time $t + \Delta t$ with limited phase error, provided the parameters are accurate Gallego & Scaramuzza (2017). Practical implementations must account for timestamp quantization, sensor readout characteristics, and discretization on the pixel grid. The choice of horizon trades off accuracy and delay: longer Δt provides a clearer separation between prediction and observation times but increases phase mismatch, which degrades cancellation if the motion model is biased or if the scene departs from ideal assumptions Gallego et al. (2018).

2.5.2 Per-event cancellation vs. batch compensation

From an architectural perspective, forward prediction at the per-event level complements batch compensation. Batch methods accumulate events, evaluate a global objective (e.g., IWE sharpness), and output motion parameters or aligned reconstructions Bardow et al. (2016); Gallego et al. (2018). In contrast, forward cancellation uses a motion model to suppress events *online*. The potential advantage is reduced downstream load and preserved latency; the risk is over-cancellation (removing informative events) if the model is inaccurate or the tolerances are too loose. The design space therefore includes prediction horizon, matching tolerances, polarity handling, and robustness to parameter bias—all explored in our sensitivity analysis.

2.5.3 Connections to bio-inspired predictive suppression

Event cameras were inspired by biological vision, where predictive coding and inhibitory mechanisms suppress predictable input to conserve bandwidth and emphasize novelty. While detailed neural models lie outside the scope of this thesis, there is a conceptual parallel: if ego-motion is predictable, the visual system can attenuate its effects to focus on behaviorally relevant changes. In event vision, recent works have discussed or implemented variants of predictive suppression via motion compensation, temporal filtering, and learned predictors Gallego et al. (2020). Our anti-event mechanism can be viewed as an explicit engineering instantiation of that principle at the sensor-processing level, tailored to circular motion for tractability.

2.5.4 Learning-based predictors (context)

Learning-based prediction methods (e.g., recurrent or spiking networks) have been proposed to extrapolate event streams or reconstruct intensity videos from events, often benefiting from large datasets and supervision or self-supervision Rebecq et al. (2019); Gallego et al. (2020). These methods capture complex priors but typically operate on event windows or voxel grids, which reintroduces batching and latency. In contrast, our focus remains on a compact analytic predictor designed for per-event operation, trading generality for speed and interpretability.

2.6 Metrics and Evaluation Practices

Evaluating event-based algorithms requires metrics that reflect both the asynchronous nature of the data and the task-specific objectives. For motion estimation and compensation, the predominant practice is to quantify the sharpness or contrast of images of warped events (IWEs): if motion parameters are correct, warped events align and the IWE exhibits high contrast; if not, the image is blurred Gallego et al. (2018). SOFIE optimized a related energy and demonstrated that contrast-based objectives correlate with perceptual sharpness and motion accuracy Bardow et al. (2016). For segmentation-by-compensation, the degree to which background events sharpen under a background motion model and foreground events remain diffuse provides a natural discriminator Stoffregen et al. (2019).

2.6.1 Rates, residuals, and SNR-like measures

Beyond contrast, event-rate statistics and residual distributions provide informative diagnostics. Because event bandwidth depends on scene dynamics, reporting event rates (events/s) and densities (events/area) is standard practice Gallego et al. (2020). In denoising and compensation tasks, comparing residual events to a background baseline approximates an SNR-like improvement: a successful algorithm should reduce structured clutter relative to inherent background activity and sensor noise. In this thesis, we therefore adopt (i) **cancellation ratio** (cancelled/total events), (ii) **residual event density** on regions of interest (e.g., disc vs. background), and (iii) **radial residual profiles** to localize where cancellation fails (e.g., near the center vs. outer rim). These choices align with evaluation practices in motion compensation and event denoising, while being tailored to circular-motion analysis Bardow et al. (2016); Gallego et al. (2018); Xu et al. (2020).

2.6.2 Ablations and sensitivity analyses

Algorithmic claims are most convincing when supported by systematic ablations. In event vision, ablations typically vary window sizes, contrast objectives, regularization, and sensor parameters Gallego et al. (2018); Rebecq et al. (2017). For proactive cancellation, the critical dimensions are prediction horizon Δt, spatial tolerance ϵ_{xy}, temporal tolerance ϵ_t, polarity handling, and motion-estimation bias. We therefore structure Chapter 8 as a sensitivity analysis: cancellation curves vs. Δt, residual maps vs. tolerances, and robustness to biased (c_x, c_y, ω). This format also clarifies trade-offs: larger tolerances increase matches (and cancellation) but risk over-cancellation; longer Δt increases phase error; strict polarity reduces false matches but may under-cancel in the presence of polarity asymmetries Gallego et al. (2020); Delbrück et al. (2020).

2.6.3 Bridging to the rest of the thesis

Finally, we emphasize that these metrics are not ends in themselves; they are proxies for downstream utility. Reducing ego-motion clutter should make it easier for subsequent modules (flow estimation, VO/VIO/SLAM, detection) to operate on informative events. While full system-level integration is beyond the scope of this thesis, our evaluation choices are designed to be translatable: a higher cancellation ratio with a lower residual density on the disc region (relative to background activity) suggests a leaner, more informative event stream for any downstream consumer Gallego et al. (2020). Chapter 4 formalizes these metrics; Chapters 8–7 quantify them empirically under rotational motion.

Chapter 3

Motion Estimation & Event Prediction

3.1 Purpose and Role in the Pipeline

Accurate ego-motion estimation is the prerequisite for our per-event forward prediction and cancellation scheme (Chapter 5). Since event cameras asynchronously report brightness changes along moving edges, the vast majority of events in a static scene under camera motion are caused by *ego-motion* Gallego et al. (2020). Estimating the rotational motion parameters allows us to predict the short-horizon trajectories of events and issue anti-events when a causal spatio-temporal gate is satisfied (Chapter 4). In this chapter we: (i) formalize the rotational model used for prediction; (ii) review estimation strategies from the literature; (iii) present our practical estimation pipeline tailored to a circular-motion rig; and (iv) analyze error sources and their impact on downstream cancellation.

3.2 Rotational Motion Model for Short Horizons

We model the dominant motion as planar rotation of the image around a center $c = (c_x, c_y)^\top$ with angular velocity ω (signed). Let $x = (x, y)^\top$ be a pixel coordinate (in pixels). The rotation operator about c by angle θ is

$$\mathcal{R}(x; c, \theta) \;=\; c + R(\theta)\,(x - c), \qquad R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{3.1}$$

Over a short prediction horizon Δt, assuming constant angular velocity, the forward-predicted location of an event occurring at time t and position x is

$$x' \;=\; \mathcal{R}(x; c, \omega\,\Delta t)\,. \tag{3.2}$$

Equation (3.2) is the geometric backbone of the predict–wait–match rule in Chapter 4. Rotation-only models are widely used in event-based motion estimation and yield closed-form sensitivities, making them attractive for controlled rigs (e.g., spinning discs) and short-horizon prediction Gallego & Scaramuzza (2017); Gallego et al. (2018); Stoffregen et al. (2019).

Normalized vs. pixel coordinates. If significant lens distortion or strong perspective effects are present, coordinates should be undistorted and normalized before applying (3.1); in our setup, we work on calibrated or approximately linearized pixel coordinates, consistent with controlled planar rigs Scheerlinck (2021); Wang (2025).

3.3 From Events to Rotation Parameters

3.3.1 Problem Statement

Given a stream of events $E = \{(x_i, t_i, p_i)\}$ from a static scene under ego-rotation, estimate (c, ω) as functions of time. Since we deploy predictions at the per-event timescale, we require estimates (or smoothed trajectories) that can be *interpolated at arbitrary timestamps.*

3.3.2 Estimation Strategy

We use *circle fitting with angle differencing* as our motion-estimation method for the circular-motion setting.

Direct angular velocity via contrast maximization

A rotational specialization of contrast maximization warps events by a hypothesized angular velocity ω (and center c) to a common reference time and seeks the ω that maximizes IWE sharpness Gallego & Scaramuzza (2017); Gallego et al. (2018). Let x_i be rotated backward from t_i to t_0 by angle $\omega\,(t_i - t_0)$ about c, producing $x_i(t_0)$; then the IWE is $I(x; t_0) = \sum_i \kappa(x - x_i(t_0))\,\sigma_i$, with $\kappa(\cdot)$ a smoothing kernel and $\sigma_i \in \{\pm 1\}$ the polarity. Common objectives are variance or contrast of I. Optimization can be performed over short packets (e.g., a few milliseconds), yielding a quasi-instantaneous $\hat{\omega}$ Gallego et al. (2018). This approach is robust to noise at adequate event densities and ties directly to rotational flow.

Circle fitting and angle-differencing

When a high-contrast rim or marker on a spinning disc produces coherent event clusters, we may fit circles to spatial event clouds to estimate c and radial trajectories. Consider a set of event coordinates $\{x_k\}$ accumulated over a very short window ΔT (small enough for negligible rotation of c). Define the algebraic circle fit objective

$$\min_{c,r} \sum_k \left(\|x_k - c\|_2 - r\right)^2, \tag{3.3}$$

which admits stable solutions via Pratt/Taubin variants or RANSAC for outlier rejection (the specific variant is implementation-dependent). With c estimated, each event location x_k yields a *bearing angle* $\theta_k = \operatorname{atan2}(y_k - c_y,\ x_k - c_x)$. Sorting by timestamps, a finite-difference estimate of angular velocity is

$$\hat{\omega}(t_k) \;\approx\; \frac{\operatorname{unwrap}(\theta_k - \theta_{k-1})}{t_k - t_{k-1}}, \tag{3.4}$$

optionally smoothed by a causal filter (e.g., exponential moving average). This procedure exploits the geometric structure of circular motion and avoids packet-wise optimization, at the cost of sensitivity to center bias and sparse sampling.

3.3.3 Interpolation and Time Alignment

Our predictor requires $(\hat{c}(t), \hat{\omega}(t))$ at the *exact* event times t_i. We therefore store time series $\{\hat{c}(t_m), \hat{\omega}(t_m)\}$ at estimation timestamps $\{t_m\}$ and evaluate them at arbitrary t by 1D interpolation:

$$\hat{c}(t) = \operatorname{interp1}(t; \{t_m\}, \{\hat{c}(t_m)\}), \quad \hat{\omega}(t) = \operatorname{interp1}(t; \{t_m\}, \{\hat{\omega}(t_m)\}). \tag{3.5}$$

In practice, linear interpolation with edge hold is sufficient given our sub-millisecond horizons and smoothly varying $\omega(t)$ on the rig. When available, an external tracker (e.g., blob tracker or encoder) can provide $\{(t_m, \hat{c}, \hat{\omega})\}$ directly, which we then interpolate at event timestamps. An asynchronous event-blob tracker can also supply high-rate motion cues Wang et al. (2024). Using IMU gyroscope as a *feed-forward* prior on ω is also common in event tracking Rebecq et al. (2017); Wang (2025).

3.4 Event Prediction under Rotation

Given an event $e_i = (x_i, t_i, p_i)$, the forward prediction for a horizon Δt applies (3.2) with the *time-aligned* parameters:

$$x'_i \;=\; \mathcal{R}(x_i;\ \hat{c}(t_i),\ \hat{\omega}(t_i)\,\Delta t), \qquad t'_i \;=\; t_i + \Delta t, \tag{3.6}$$

and we adopt the predicted tuple

$$e'_i = (x'_i,\ t'_i,\ p'_i), \quad p'_i = -p_i,$$

matching the notation in Chapter 4. These predicted events feed the spatio-temporal gate. Because Δt is short (typically $\leq$ a few milliseconds), assuming piecewise-constant ω over $[t_i, t_i + \Delta t]$ is reasonable for rigs with slowly varying spin rate Gallego & Scaramuzza (2017); Scheerlinck (2021).

3.5 Error Sources and Sensitivity (pointer)

Error terms and gate-sizing are detailed once in Chapter 4 (Eqs. (4.7)–(4.9)). Here we refer to that analysis to avoid repetition.

3.6 Our Practical Estimation Pipeline

We adopt a pragmatic pipeline tailored to the disc rig and asynchronous cancellation:

(1) Event collection and short-window accumulation

Events are processed in short, overlapping windows (e.g., a few milliseconds) to form sparse spatial clouds suitable for circle fitting. Window length balances spatial support (too small $\Rightarrow$ sparse) and motion stationarity (too large $\Rightarrow$ parameter drift).

(2) Center estimation via circle fitting

We solve (3.3) on the accumulated cloud using a robust algebraic fit (Pratt/Taubin) with optional RANSAC to mitigate outliers from background activity and edge fragments. The resulting $\hat{c}$ is smoothed over time with a causal low-pass filter.

(3) Angular velocity estimation

We use *angle differencing*: compute angles θ_k around $\hat{c}$ and evaluate (3.4) with causal smoothing. This is lightweight and well-matched to strong circular structure. When available, an external tracker or gyroscope prior provides $\tilde{\omega}$, which we fuse by exponential averaging:

$$\hat{\omega}(t_k) \leftarrow (1-\alpha)\,\hat{\omega}(t_{k^-}) + \alpha\,\tilde{\omega}(t_k), \qquad \alpha \in (0,1], \tag{3.7}$$

trading responsiveness for noise attenuation Rebecq et al. (2017).

(4) Interpolation to event times

We store $\{\hat{c}(t_m), \hat{\omega}(t_m)\}$ and evaluate (3.5) at *every* event time t_i before applying (3.6). This preserves causality and avoids batching, aligning with our per-event cancellation design.

(5) Sanity checks and fast diagnostics

As an online diagnostic, we monitor a *radial residual profile* (after cancellation) to detect center bias (flat offset) and angular-velocity drift (radius-dependent widening), mirroring alignment criteria in segmentation-by-compensation Stoffregen et al. (2019).

3.7 Validation and Bridge to Experiments

We validate $(\hat{c}, \hat{\omega})$ in two ways. First, when an external tracker is used, we compare $\hat{\omega}$ to the tracker's angular velocity (or to a derived encoder ground truth) and report absolute and relative errors over time. Second, we assess *functional* quality via downstream metrics: cancellation ratio, residual density on background vs. disc masks, and radial residual profiles (Chapter 8). Improvements in these metrics under consistent gating settings indicate better short-horizon alignment, consistent with event-warping principles Gallego et al. (2018).

3.8 Discussion and Limitations

Rotation-only modeling simplifies short-horizon prediction and supports causal operation, but suffers when translation is non-negligible or the rotation center drifts. In principle, handling translation requires either scene depth (or inverse depth) or restrictive assumptions (e.g., far-scene, planar scene) Gallego et al. (2020, 2018); Rebecq et al. (2017); Benosman et al. (2014). We therefore did not include translation in our implementation; on our rig and time horizons, rotation dominates the image motion. A possible extension is to include a small, regularized translation component under a far-scene/constant inverse-depth assumption as a prior ("weak translation"), but we leave this as future work Gallego et al. (2020). Gyroscope fusion can also stabilize $\hat{\omega}$ and reduce phase error Rebecq et al. (2017). Learning-based estimators could supply priors for $\hat{c}$ or refine $\hat{\omega}$, but would likely introduce latency and dependence on training distributions Zhu et al. (2018, 2019). Our emphasis remains on lightweight, analyzable estimation that integrates cleanly with per-event cancellation.

Summary. This chapter presented a rotation-only motion model for short-horizon prediction, reviewed estimation strategies with events, and described a practical, causal pipeline to produce time-aligned $(\hat{c}, \hat{\omega})$ for per-event forward prediction. The next chapter details the cancellation rule and implementation, and Chapter 8 quantifies sensitivity to Δt, spatial/temporal tolerances, and parameter biases.

Chapter 4

Problem Formulation

4.1 Notation and Event Stream Model

Let $\Omega \subset \mathbb{Z}^2$ denote the pixel grid and $t \in \mathbb{R}_{\geq 0}$ denote continuous time. An event is the tuple

$$e_i := (x_i, y_i, t_i, p_i) \in \Omega \times \mathbb{R}_{\geq 0} \times \{-1, +1\},$$

with polarity p_i indicating the sign of the log-intensity change. The event stream over an interval $[0, T]$ is the multiset $E := \{e_i\}_{i=1}^{N(T)}$, where $N(T)$ denotes the number of events observed in $[0, T]$.

We adopt the standard log-intensity threshold model of event generation: a pixel triggers an event when the change in log-intensity exceeds a (per-pixel) contrast threshold c; see Lichtsteiner et al. (2008); Brandli et al. (2014); Posch et al. (2014); Gallego et al. (2020). Following common practice, the stream can be represented as a distribution of impulses at timestamps and locations. Let $u \in \Omega$ denote a pixel coordinate (to avoid conflict with the polarity symbol p), and let $\sigma_i := p_i \in \{-1, +1\}$ denote the event polarity. Then an event field $E(u, t)$ integrates Dirac deltas whose amplitudes encode polarity and quantized contrast Scheerlinck (2021); Wang (2025):

$$E(u, t) = \sum_i \sigma_i \, c \, \delta(t - t_i) \; \mathbf{1}\{u = (x_i, y_i)\}, \qquad \sigma_i \in \{-1, +1\}. \tag{4.1}$$

Per-pixel non-idealities (background activity, polarity asymmetry, threshold mismatch, refractory effects) are acknowledged and handled operationally through robust matching tolerances and polarity checks introduced below Brandli et al. (2014); Delbrück et al. (2020); Gallego et al. (2020). Timestamp jitter on modern devices is typically small relative to other uncertainties at sub-millisecond horizons Wang (2025).

Objective. Consistent with the high-level research questions in the introduction, our objective is to identify and *cancel* events that are predictable from a circular-motion model, *asynchronously and per-event.* The aim is to reduce ego-motion clutter early, preserving low latency while leaving residual events that more likely correspond to independent scene/object motion.

4.2 Circular-Motion Ego-Motion Model

Over short horizons, we model the dominant camera motion as planar rotation about an image-plane center $c = (c_x, c_y) \in \mathbb{R}^2$ with signed angular velocity $\omega \in \mathbb{R}$. For $x \in \mathbb{R}^2$, define the rotation of x about c by angle θ as

$$\mathcal{R}(x; c, \theta) \;=\; c \;+\; R(\theta)\,(x - c), \qquad R(\theta) \;=\; \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{4.2}$$

Assuming ω locally constant over a short prediction horizon Δt, the *future* location of x is

$$x' \;=\; \mathcal{R}(x;\, c, \omega\,\Delta t)\,. \tag{4.3}$$

For an event $e_i = (x_i, y_i, t_i, p_i)$ at time t_i, write the pixel coordinate as $x_i = (x_i, y_i)$ and denote its forward-predicted location at time $t_i + \Delta t$ as $x_i' = \mathcal{R}(x_i; c, \omega\,\Delta t)$.

Rationale and scope. The rotational model is useful beyond the specific rig. (i) It is *depth-free and identifiable*: (c, ω) is observable from events alone, whereas cancellation with translation generally requires scene depth (or inverse depth) and a richer motion field. (ii) Rotation is a dominant component of hand-held and robotic ego-motion at moderate distances; over short horizons the parallax due to translation is second order relative to the angular term. (iii) The model is analytically compact (two parameters) and widely used for stress-testing event pipelines Gallego & Scaramuzza (2017); Stoffregen et al. (2019); Gallego et al. (2018). Our goal is to establish per-event cancellation under a well-posed, low-latency model. Extensions can incorporate constant translational velocity together with an inverse-depth parameterization; we discuss this path and trade-offs in §4.7 and Chapter 8.

4.3 Per-Event Forward Prediction with a Temporal Gate

We adopt a causal *predict–wait–match* strategy with an explicit temporal tolerance:

1. **Predict.** Upon observing e_i at t_i, form the *predicted event*

$$e_i' = (x_i',\, t_i',\, p_i'), \quad x_i' = \mathcal{R}(x_i; c, \omega\,\Delta t),\; t_i' = t_i + \Delta t,\; p_i' = -p_i,$$

   i.e., the expected location, time and opposite polarity of the cancellable counterpart.

2. **Wait.** Defer the decision until the decision time $t^* = t_i'$.

3. **Temporal gate.** At t^*, compare the single predicted event e_i' against *real* events

whose timestamps lie inside a symmetric window:

$$\mathcal{N}_t(t^*;\epsilon_t) \;=\; \{\, e_j = (x_j, y_j, t_j, p_j) \in E \;:\; |t_j - t^*| \le \epsilon_t \,\}. \tag{4.4}$$

4. **Spatial/polarity gate.** Within $\mathcal{N}_t(t^*;\epsilon_t)$, search for a real event e_j satisfying

$$\|x_j - x_i'\|_2 \le \epsilon_{xy}, \qquad \pi(p_i, p_j) = 1, \tag{4.5}$$

   where $\epsilon_{xy} > 0$ is a spatial tolerance and π encodes the polarity rule (default: strict opposite polarity, $\pi(p_i, p_j) = \mathbf{1}\{p_j = -p_i\}$).

If such a match exists, declare e_i *ego-motion predictable* and add e_i to the cancel set; remove both e_i and the matched event e_j from the stream (software-level cancellation). If no match is found inside the spatio-temporal gate, e_i remains in the residual set.

Remarks on implementation. Equation (4.4) enforces a temporal gate with tolerance ϵ_t (no fixed time bins). In practice, we maintain two time-sorted buffers and advance sliding indices so that, for each t^*, we retrieve exactly those real events with $|t - t^*| \le \epsilon_t$ before applying the spatial/polarity test (4.5). If (c, ω) are provided at discrete tracker timestamps, we evaluate $\mathcal{R}(\cdot)$ using parameters interpolated to t_i and t^* (linear or spline), ensuring consistent predictions. The cancellation removes matched pairs from the event stream, preserving causality and low latency without hardware-level interference.

Discussion of polarity. Over short intervals, a moving edge driving monotonic log-intensity change tends to produce consistent polarity. We therefore default to *opposite-polarity cancellation* (matching predicted opposite polarity $\bar{p}_i$ with the real event polarity). When polarity asymmetries are suspected (sensor mismatch), we also evaluate a relaxed polarity policy in sensitivity analyses.

4.3.1 Matching Policy and One-to-One Constraints

Let $\mathcal{N}_t(t^*;\epsilon_t)$ denote the candidate window at decision time. We impose a *one-to-one* pairing between predicted and real events within that window to prevent over-cancellation. A simple and effective policy is *mutual nearest neighbors* within the spatial tolerance: for each real event, find its nearest predicted candidate and vice versa; accept a pair only if both choose each other and (4.5) holds (polarity included). Greedy distance-ordered matching with a "used set" for predicted events is an alternative with similar behavior.

4.3.2 Sets, Counters, and Residuals

Over $[0, T]$, let E denote all observed events, $\mathcal{C} \subseteq E$ the subset declared cancellable under (4.4)–(4.5) and removed via software-level cancellation, and $\mathcal{R} := E \backslash \mathcal{C}$ the *residual* events. Define the *cancellation ratio*

$$\mathrm{CR} \ := \ \frac{|\mathcal{C}|}{|E|}, \qquad \mathrm{RR}_{\mathcal{A}} \ := \ \frac{|\mathcal{R} \cap \mathcal{A}|}{|\mathcal{A}|} \tag{4.6}$$

for a region-of-interest $\mathcal{A} \subseteq \Omega \times [0, T]$ (e.g., disc vs. background masks). We also report *residual event density* $\rho_{\mathcal{A}} := \frac{|\mathcal{R} \cap (\Omega_{\mathcal{A}} \times [0,T])|}{|\Omega_{\mathcal{A}}|}$ and radial residual profiles on the disc (§4.8).

4.3.3 Causality, Buffers, and Complexity

The per-event pipeline maintains short spatio-temporal buffers: (i) a small queue of pending predictions indexed by decision times t^*; (ii) a time-sorted buffer of real events for the sliding temporal window (4.4); and (iii) a light spatial index (e.g., grid or k-d tree) for neighborhood queries within radius ϵ_{xy}. All matches are decided using only past and present data at the time of processing (no future timestamps are consulted), preserving causality and low latency (no global batch optimization), similar in spirit to asynchronous processing emphasized in Wang (2025); Scheerlinck (2021).

4.4 Geometric and Temporal Sensitivity to Model Errors

Prediction accuracy depends on parameter bias and timing. Let the true parameters be $(c^\star, \omega^\star)$ and the estimates be $(\hat{c}, \hat{\omega})$. Consider an event at radius $r = \|x - c^\star\|_2$ from the true center.

Angular-velocity bias. Let $\Delta\omega := \hat{\omega} - \omega^\star$. Over horizon Δt, the true angular displacement is $\theta^\star = \omega^\star \Delta t$ and the predicted is $\hat{\theta} = \hat{\omega}\Delta t$. The angular error $\delta\theta = \hat{\theta} - \theta^\star = (\Delta\omega)\Delta t$ induces a spatial prediction error

$$\varepsilon_\omega(r, \Delta t) \ = \ \left\|\mathcal{R}(x; c^\star, \hat{\theta}) - \mathcal{R}(x; c^\star, \theta^\star)\right\| \ = \ 2r \left|\sin(\delta\theta/2)\right| \ \approx \ r\,|\Delta\omega|\,\Delta t, \tag{4.7}$$

where the linear approximation holds for small $\delta\theta$.

Center-of-rotation bias. Let $\Delta c := \hat{c} - c^\star$. For small Δt, the first-order contribution of Δc to the spatial error satisfies

$$\varepsilon_c(r) \ \lesssim \ \|\Delta c\|_2, \tag{4.8}$$

and couples with (4.7) for general r. This bound follows from a first-order expansion of the rotation about a perturbed center: for fixed angular displacement the mapping $x \mapsto c+R(\theta)(x-c)$ is Lipschitz in c, hence miscentering by Δc produces at most $\mathcal{O}(\|\Delta c\|_2)$ displacement (triangle inequality), independent of r to first order.

Timing uncertainty, horizon, and the temporal gate. To avoid confusion with the polarity symbol σ_i used in (4.1), we denote effective timing uncertainty by τ_t (sensor latency variability, timestamp quantization; see hardware timing notes in Lichtsteiner et al. (2008); Delbruck et al. (2014)). Here, "phase" refers to the angular displacement $\theta = \omega\,\Delta t$. Over horizon Δt, the phase uncertainty is $\sigma_\theta \approx |\omega^\star|\,\tau_t$, contributing spatial uncertainty $\sigma_x \approx r\,\sigma_\theta \approx r\,|\omega^\star|\,\tau_t$. The temporal gate (4.4) explicitly allows $|t_j - t^*| \leq \epsilon_t$ so that modest timing error does not cause missed matches. However, increasing Δt amplifies phase error via (4.7); thus Δt trades discriminability (separation in time) against phase robustness.

Implications for the spatial gate. A sufficient condition for a match is

$$\varepsilon_\omega(r, \Delta t) + \varepsilon_c(r) + \sigma_x \;\leq\; \epsilon_{xy}, \tag{4.9}$$

which explains typical cancellation curves: CR increases with ϵ_{xy} but saturates/degrades if ϵ_{xy} is too large (false pairs); and CR decreases with large Δt as phase error grows. These trends motivate the parameter sweeps in Chapter 8.

4.5 Formal Cancellation Rule

Define the decision indicator

$$\delta(e_i; \hat{c}, \hat{\omega}, \Delta t, \epsilon_{xy}, \epsilon_t) \;=\; \begin{cases} 1, & \text{if } \exists\, e_j \in \mathcal{N}_t(t_i + \Delta t; \epsilon_t) \text{ s.t. (4.5) holds,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

Then $\mathcal{C} = \{e_i \in E : \delta(e_i; \cdot) = 1\}$ and $\mathcal{R} = E \backslash \mathcal{C}$. Rather than posing a global optimization, we study the *sensitivity* of cancellation outcomes to $(\Delta t, \epsilon_{xy}, \epsilon_t)$ given $(\hat{c}, \hat{\omega})$ in Chapter 8, and we report residuals to guard against over-cancellation.

4.6 Cancellation Implementation and Representation

On a successful match, we remove both the originating event e_i and the matched event e_j from the event stream, effectively suppressing the predicted observation in the maintained event representation (e.g., polarity-separated accumulation or voxel grid). This software-level cancellation operates at the decision time t^* with microsecond-scale delay, consistent with asynchronous processing goals Gallego et al. (2018); Bardow et al. (2016).

In evaluation, matched pairs are simply omitted from downstream processing, avoiding the need for explicit "anti-event" data structures.

4.7 Assumptions, Limitations, and Threats to Validity

(A1) Static scene. The background is static; independently moving objects are not modeled by the circular ego-motion and should remain as residuals.

(A2) Short-horizon rotation dominance. Over small Δt, rotational flow dominates translation for our apparatus; accelerations are small over the horizon. This aligns with rotation-focused practice in event tracking and compensation Gallego & Scaramuzza (2017); Gallego et al. (2018).

(A3) Calibration and warping. Intrinsics and distortion are pre-compensated or negligible on the ROI; otherwise, normalize coordinates before applying (4.3).

(A4) Sensor non-idealities. Background activity, threshold mismatch, polarity asymmetry, and refractory effects are present Brandli et al. (2014); Delbrück et al. (2020); Gallego et al. (2020). We mitigate via strict polarity checks, tuned tolerances $(\epsilon_{xy}, \epsilon_t)$, and background baselines in the metrics. Timestamp jitter is comparatively minor at our horizons Wang (2025).

(A5) Causality and bounded memory. Decisions use only past/present data in short buffers; we avoid global contrast-maximization windows and maintain per-event latency Bardow et al. (2016); Gallego et al. (2018).

Failure modes. (i) Larger Δt and/or angular-velocity bias $\Delta\omega$ cause phase errors (4.7) and missed matches; (ii) miscentered rotation Δc yields radius-dependent residuals; (iii) overly large tolerances over-cancel by matching unrelated events; (iv) flicker can pass spatio-temporal gates unless filtered; (v) pronounced translation or non-circular motion violates the model.

4.8 Bridge to Metrics and Experiments

We quantify performance via: (i) cancellation ratio CR (4.6); (ii) residual density $\rho_{\mathcal{A}}$ on disc vs. background masks; (iii) radial residual profiles on the disc; and (iv) sensitivity curves vs. Δt, ϵ_{xy}, ϵ_t, polarity handling, and parameter bias (Chapter 8). These choices reflect established motion-compensation practice (IWE/contrast) and event-rate diagnostics Bardow et al. (2016); Gallego et al. (2018); Stoffregen et al. (2019); Gallego et al. (2020), adapted to proactive per-event cancellation.

Chapter 5

Event Cancellation Algorithm

5.1 Overview and Design Goals

The cancellation stage implements the *predict–wait–match* rule from Chapter 4: for each observed event $e_i = (x_i, t_i, p_i)$, we predict a future location x'_i under rotation about $\hat{c}(t_i)$ with angular velocity $\hat{\omega}(t_i)$ over a short horizon Δt; at decision time $t'_i = t_i + \Delta t$, we search for a real event $e_j = (x_j, t_j, p_j)$ within a *temporal gate* $|t_j - t'_i| \leq \epsilon_t$ and a spatial gate $\|x_j - x'_i\|_2 \leq \epsilon_{xy}$, with polarity consistency. Matches are canceled (suppressed) and unmatched events remain as residuals. The algorithm is strictly *causal*, per-event, and parameterized by $(\Delta t, \epsilon_t, \epsilon_{xy})$, connecting directly to the sensitivity analysis in Chapter 8. Compared to batch motion compensation/contrast maximization Gallego et al. (2018); Stoffregen et al. (2019); Bardow et al. (2016), our design trades some global robustness for low latency and interpretability.

5.2 Inputs, Outputs, and State

Inputs.

- Event stream $E = \{(x_i, t_i, p_i)\}$, time-sorted (polarity $p_i \in \{-1, +1\}$).
- Time-aligned motion parameters $(\hat{c}(t), \hat{\omega}(t))$ from Chapter 3.
- Hyperparameters: prediction horizon Δt, temporal tolerance ϵ_t, spatial tolerance ϵ_{xy}, polarity mode π (strict opposite, equal, or ignore).

Outputs.

- Canceled set $\mathcal{C} \subseteq E$ and residual set $\mathcal{R} = E \setminus \mathcal{C}$.
- The cancellation is implemented by dropping matched pairs from the event stream; no explicit "anti-event" data structures are maintained.

State.

- A time-sorted buffer of *recent real events* for window queries.

- A time-sorted buffer (queue) of *pending predicted locations and decision times* keyed by $t'_i = t_i + \Delta t$.
- A light spatial index (e.g., KD-tree or uniform grid) over candidates within the current temporal window.

5.3 Per-Event Prediction

For each incoming real event $e_i = (x_i, t_i, p_i)$, we compute the predicted tuple

$$x'_i \;=\; \mathcal{R}(x_i;\, \hat{c}(t_i),\, \hat{\omega}(t_i)\,\Delta t)\,, \qquad t'_i \;=\; t_i + \Delta t, \qquad \bar{p}_i \;=\; -p_i, \tag{5.1}$$

and push $\hat{e}_i = (x'_i, t'_i, \bar{p}_i)$ into a min-heap keyed by t'_i.[1] Time-aligned $(\hat{c}, \hat{\omega})$ are obtained by interpolation (3.5).

5.4 Temporal Gate and Candidate Retrieval

At the current event time τ, pop all predictions with $t'_i \le \tau$. For each popped $\hat{e}_i$, form the candidate set

$$\mathcal{N}_t(t'_i; \epsilon_t) \;=\; \{\, e_j = (x_j, t_j, p_j) \in E \mid |t_j - t'_i| \le \epsilon_t \,\}, \tag{5.2}$$

using two sliding indices over the time-sorted real-event buffer to avoid re-scanning the stream. This realizes the temporal gate introduced in Chapter 4 (§4.3) without fixed-width bins, thereby suppressing boundary artifacts that otherwise degrade matching.

5.5 Spatial & Polarity Gating, and Pairing Policy

Within $\mathcal{N}_t(t'_i; \epsilon_t)$, we enforce

$$\|x_j - x'_i\|_2 \le \epsilon_{xy}, \qquad \pi(p_i, p_j) = 1, \tag{5.3}$$

where π encodes the polarity rule. By default we use *strict opposite* ($p_j = -p_i$), which minimizes over-cancellation on monotonic edges; we also evaluate *ignore* (no polarity test) and *equal* (for ablations) in Chapter 8 (cf. predictive suppression analogies in Hosoya et al. (2005); Rao & Ballard (1999)).

Sizing the gates. We choose $(\epsilon_t, \epsilon_{xy})$ so that the total prediction error remains within the

[1] We use pixel-space rotation (5.1). If significant distortion is present, apply the mapping in normalized coordinates (Chapter 3).

spatial gate (see Chapter 4; cf. Gallego et al. (2018); Xu et al. (2020)):

$$\varepsilon_\omega(r, \Delta t) + \varepsilon_c(r) + \sigma_x \;\leq\; \epsilon_{xy}, \tag{5.4}$$

where ε_ω captures angular-velocity bias over horizon Δt, ε_c center bias, and σ_x timing-induced spatial uncertainty. A practical guide for linking temporal and spatial tolerances is $\epsilon_t \approx \epsilon_{xy}/(r\,|\omega|)$ (see the temporal-tolerance study in Chapter 8; also consistent with motion-compensation time–space trade-offs in Gallego et al. (2018); Xu et al. (2020)).

Mutual nearest neighbors (MNN). To prevent many-to-one matches in dense bursts, we adopt an MNN policy within the gate: let $\mathcal{P}$ be the set of popped predictions in $[t'_i - \epsilon_t,\, t'_i + \epsilon_t]$, and $\mathcal{R}_t$ the real events in that same window. Build a KD-tree on $\mathcal{P}$ positions $\{x'_i\}$ and query nearest neighbors for each x_j in $\mathcal{R}_t$ (within ϵ_{xy}), and symmetrically a KD-tree on $\mathcal{R}_t$ queried by $\{x'_i\}$. Accept a pair $(\hat{e}_i, e_j)$ if and only if they select each other and pass (5.3). This is a standard trick to increase precision in nearest-neighbor association while keeping complexity near-linear in the number of candidates.

Adaptive spatial tolerance (optional). Empirically, spatial error grows with radius and timing uncertainty (Chapter 4). A practical heuristic is

$$\epsilon_{xy}(r) \;=\; k\left(r\,|\hat{\omega}(t_i)|\,\tau_t \;+\; \|\Delta c\|\right), \quad r = \|x_i - \hat{c}(t_i)\|_2, \quad k \in [1.5, 2.5], \tag{5.5}$$

with τ_t the effective timestamp uncertainty. We report both fixed and adaptive tolerances in sensitivity experiments.

5.6 Cancellation and Residuals

For each accepted pair $(\hat{e}_i, e_j)$, we:

1. Mark $e_i \in \mathcal{C}$ (the *cause* event) and remove the matched event e_j from the maintained event representation (e.g., polarity-separated accumulator or voxel grid).

2. Software-wise, dropping both e_i and e_j from the stream is equivalent to perfect cancellation under our signed rasterization.

All unmatched events remain in the residual set and form the high-pass visualization used in qualitative figures. This mirrors the subtraction principle behind motion-compensated alignment in warping-based methods Gallego et al. (2018); Stoffregen et al. (2019) but acts *proactively* at the event time scale.

5.7 Complexity and Throughput

Let n_w be the number of events in a typical temporal window of width $2\epsilon_t$. Building a KD-tree per window costs $\mathcal{O}(n_w \log n_w)$ and querying is $\mathcal{O}(n_w \log n_w)$, but in practice we reuse structures by batching popped predictions with similar t' or by using uniform spatial grids for near-constant-time neighborhood queries. Since windows are short and sparsity is high, the dominant cost is memory traffic; the implementation therefore uses contiguous arrays, time-sorted buffers, and chunked processing to achieve high throughput.

5.8 Numerical Details and Edge Cases

Sub-pixel rasterization. We visualize residuals with signed bilinear splatting (Chapter 8), which stabilizes appearances across sub-pixel positions and avoids aliasing common to nearest-neighbor binning.

Time alignment and latency. All decisions are taken using only past and present data at the time of processing (no future timestamps are consulted). In practice, predictions with $|t'_k - \tau| \leq \epsilon_t$ are popped and matched to real events already observed up to τ. The microsecond-scale delay is dominated by data structure updates. Cancellation operates purely in software by removing matched pairs from the stream, with no hardware-level event emission required.

Polarity asymmetry. Real sensors can exhibit polarity-dependent thresholds; strict opposite-polarity matching is conservative. We therefore report both strict and relaxed polarity modes to expose potential asymmetry.

Boundary conditions. Predictions that fall outside the image plane (after (5.1)) are discarded. For safety, we clip coordinates prior to spatial indexing to avoid invalid memory accesses.

5.9 Algorithm

Algorithm 1 Causal per-event cancellation with temporal gate

```
Require: time-sorted events $E = \{(x_i, y_i, p_i, t_i)\}_{i=1}^{N}$, parameters $(\hat{\mathbf{c}}(t), \hat{\omega}(t))$, horizon $\Delta t$, tolerances $(\varepsilon_t, \varepsilon_{xy})$, polarity mode $\Pi$
PredQ $\leftarrow$ empty min-heap keyed by predicted time $t'$
RealBuf $\leftarrow$ empty deque;    $L \leftarrow 1,\ R \leftarrow 1$
for each event $e_i = (x_i, y_i, p_i, t_i) \in E$ do
    $\hat{\mathbf{c}}_i \leftarrow$ Interp$(\hat{\mathbf{c}}, t_i)$;   $\hat{\omega}_i \leftarrow$ Interp$(\hat{\omega}, t_i)$
    $(x'_i, y'_i) \leftarrow$ Rotate$((x_i, y_i); \hat{\mathbf{c}}_i, \hat{\omega}_i \cdot \Delta t)$;   $t'_i \leftarrow t_i + \Delta t$;   $\bar{p}_i \leftarrow$ FlipPol$(p_i)$
    Push$\big($PredQ$, \hat{e}_i = (x'_i, y'_i, \bar{p}_i, t'_i, i)\big)$
    Append$($RealBuf$, e_i)$
    while PredQ $\neq \varnothing$ and Top(PredQ)$.t' \leq t_i$ do
        $\mathcal{P} \leftarrow$ PopAllWithin(PredQ, $t_i$, $\varepsilon_t$)        ▷ all predictions with $|t'_k - t_i| \leq \varepsilon_t$
        advance $(L, R)$ so that RealBuf$[L{:}R]$ satisfies $|t_j - t'_k| \leq \varepsilon_t$ for some $k$
        build spatial index (e.g., KD-tree) on $\{(x'_k, y'_k) \in \mathcal{P}\}$
        $\mathcal{A} \leftarrow \varnothing$        ▷ candidate assignments: $(j, k, d)$
        for each $e_j = (x_j, y_j, p_j, t_j) \in$ RealBuf$[L{:}R]$ do
            find nearest $\hat{e}_k \in \mathcal{P}$ within $\varepsilon_{xy}$ to $(x_j, y_j)$
            if $\hat{e}_k$ exists and $\Pi(\bar{p}_k, p_j)$ holds then
                add $(j, k, \|(x_j, y_j) - (x'_k, y'_k)\|)$ to $\mathcal{A}$
            end if
        end for
        sort $\mathcal{A}$ by distance ascending; greedily accept one-to-one matches
        record accepted pairs into $\mathcal{C}$; mark matched $e_j$ and $\hat{e}_k$ as canceled
        remove (or flag) matched $e_j$ from RealBuf; discard used $\hat{e}_k$ from $\mathcal{P}$
    end while
end for
return matches $\mathcal{C}$ and residuals: real $E \setminus \{e_j : (j,k) \in \mathcal{C}\}$, predicted $\{\hat{e}_k\} \setminus \{\hat{e}_k : (j,k) \in \mathcal{C}\}$
```

5.10 Connection to Literature and Rationale

The spatial/temporal gating echoes the alignment logic of IWE-based methods Gallego et al. (2018), with the key difference that we *predict and cancel causally* rather than optimize parameters over packets. Motion segmentation by motion compensation Stoffregen et al. (2019) similarly validates motion hypotheses via alignment but uses iterative EM and global objectives; our algorithm replaces those with a local gate tuned by $(\Delta t, \epsilon_t, \epsilon_{xy})$. This allows immediate suppression of predictable ego-motion events via software-level cancellation (matched-pair removal), reducing bandwidth and computation for downstream tasks, a pragmatic alternative when low latency is paramount.

5.11 Parameter Selection Heuristics

The following guidelines are used to initialize and sweep parameters (formal sweeps appear in Chapter 8).

- **Prediction horizon Δt:** Start at 0.5 ms; sweep 0.25–2 ms. Larger Δt increases temporal separation but also phase error per (4.7).
- **Temporal tolerance ϵ_t:** Choose 2–3 ms to absorb timestamp variability; relate to sensor timing τ_t and spin rate $|\hat{\omega}|$ (Chapter 4).
- **Spatial tolerance ϵ_{xy}:** Start at 1.5–2 px; consider adaptive $\epsilon_{xy}(r)$ via (5.5) for high radii.
- **Polarity mode π:** Use strict opposite for conservative cancellation; report *ignore* as an ablation to probe polarity asymmetry.

5.12 Limitations and Failure Modes

Cancellation suffers when (i) Δc or $\Delta\omega$ are biased, yielding systematic residuals (Chapter 3); (ii) the scene violates the rotation-only model (translation/parallax); (iii) flicker or structured noise passes the gate; or (iv) the gate is too permissive, causing over-cancellation. Diagnostics such as radial residual profiles and background residual density (Chapter 8) help detect and correct these cases.

Summary. We presented a causal, per-event cancellation algorithm with a temporal gate, mutual nearest-neighbor pairing, and optional adaptive spatial tolerance. The design is lightweight and analyzable, implements software-level cancellation by removing matched pairs from the event stream, integrates directly with the rotation-only predictor, and provides interpretable knobs $(\Delta t, \epsilon_t, \epsilon_{xy})$ whose effects are quantified in the next chapter.

Chapter 6

Experimental Setup

This chapter documents the hardware apparatus, data-collection protocol, and software pipeline used to evaluate the causal, per-event cancellation method described in Chapters 4–5. Our goal is to provide enough detail for faithful reproduction: sensor and rig configuration, calibration steps, sequence design, and the end-to-end processing stack (estimation → prediction → gating → metrics).

6.1 Hardware and Apparatus

6.1.1 Event Camera and Optics

We used a single event-based sensor mounted on a rigid aluminum frame facing a rotating disc (Fig. 6.1). The relevant characteristics:

- **Sensor:** Prophesee Gen4 event camera (1280×720). Timestamp precision on the order of microseconds; polarity encoding in $\{\pm 1\}$ (or $\{0, 1\}$). Dynamic range $\sim$120 dB Finateu et al. (2020).
- **Lens:** Standard C-mount lens, focal length selected to provide adequate field of view for the disc. Focus was carefully set at the disc plane to maximize edge sharpness and contrast. Aperture chosen to avoid saturation while maintaining high contrast on the textured disc patterns.
- **Mounting:** Camera mounted on a tripod aimed at the rotating disc. The optical axis was approximately orthogonal to the disc plane over the region of interest.
- **Working distance:** Approximately 60–80 cm from sensor to disc plane (calibrated via known disc diameter and imaged radius), yielding an imaged disc radius of 264 px at the chosen focal length. This provided sufficient spatial resolution for motion estimation and cancellation analysis.

6.1.2 Spinning Disc Rig

The target is a flat wooden disc with an *uneven, high-contrast pattern* adhered to the visible face to generate rich edge structures under rotation (Fig. 6.2). Hardware elements:

- **Disc:** Circular wooden plate approximately 30–40 cm in diameter, with thickness sufficient to maintain rigidity under rotation. Six square paper patterns are affixed to the disc face using blue tape at the corners, creating a grid layout (visible in Fig. 6.1). The patterns consist of:
  - Four squares with intricate, flowing concentric black and white lines resembling topographical contours or fingerprint patterns, providing dense, curved edge structures.
  - Two squares (top corners) with dense random assortments of black and white shapes including squiggles, hearts, stars, circles, and crosses, creating high-frequency spatial content.

  These patterns generate a rich variety of edge orientations and spatial frequencies to stress-test the cancellation algorithm across different geometric configurations.
- **Actuation:** Precision DC or BLDC motor with controller; rotation rate adjustable via voltage or PWM command from the test equipment (visible on top shelf in Fig. 6.1). Nominal speeds tested: 1.5–3.5 rev/s, corresponding to angular velocities $\omega \approx 1.5 - -3.5\,\mathrm{rad/s}$, with steady-state angular velocity ripple below 2% after warm-up.
- **Spindle and bearings:** Low-runout precision bearings support the spindle to minimize lateral wobble and eccentricity. A hub mount ensures concentricity between the disc and rotation axis, maintaining stable circular motion for consistent experimental conditions.
- **Coordinate system:** A red hand-drawn coordinate system on the grey mounting board (visible in Fig. 6.1, lower-left) defines the reference frame with x pointing right and y pointing down, consistent with image-plane conventions.

6.1.3 Lighting Conditions

Event generation depends on brightness gradients and texture. We used ambient room lighting only (curtains closed to minimize external variation); no external lighting was employed. Illumination varied slightly with the room lights but was stable during each sequence.

6.1.4 Calibration and Referencing

We performed lightweight calibration sufficient for short-horizon rotation modeling:

- **Intrinsics/distortion:** No intrinsic calibration was performed. Processing was done in pixel coordinates; distortion was not explicitly compensated.

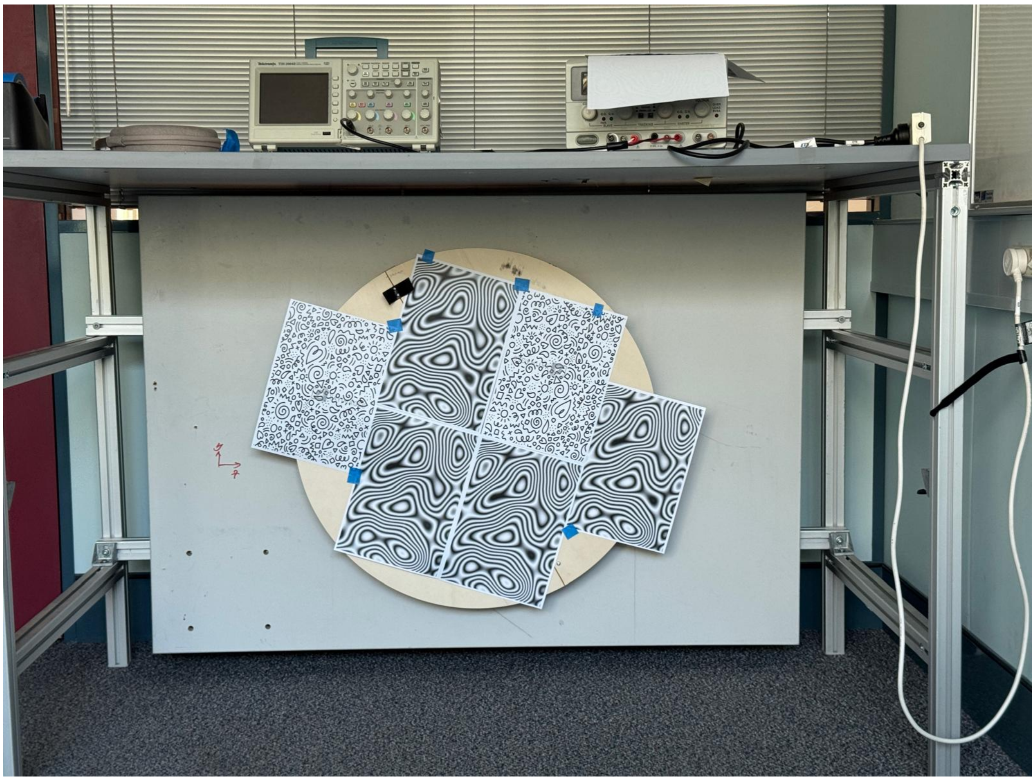

Figure 6.1: Experimental apparatus. The event camera (tripod-mounted, off-camera right) views a patterned wooden disc mounted on a rotating spindle. The disc features six high-contrast paper patterns affixed with blue tape, creating varied edge structures for rich event generation. The grey mounting board provides a stable backdrop with a hand-drawn coordinate system (red x/y axes, lower-left). Ambient room lighting was used.

- **Disc center in image:** Estimated via circle fitting on short event windows; see Chapter 3 for the method.

- **Angular velocity reference:** Estimated by angle differencing around the fitted center with causal smoothing; see Chapter 3.

6.2 Datasets and Protocol

6.2.1 Sequences

We recorded multiple *spinning-disc* sequences by varying the rotation rate and exposure to stress different motion regimes. Each sequence contains the raw, time-ordered event stream:

$$E = \left\{(x_i, y_i, p_i, t_i)\right\}_{i=1}^{N},$$

with timestamps in microseconds (converted to seconds in software). For most analyses we also used a tracker CSV with center and angular-velocity estimates; when absent, these were inferred from events via circle fit and angle differencing (Chapter 3).

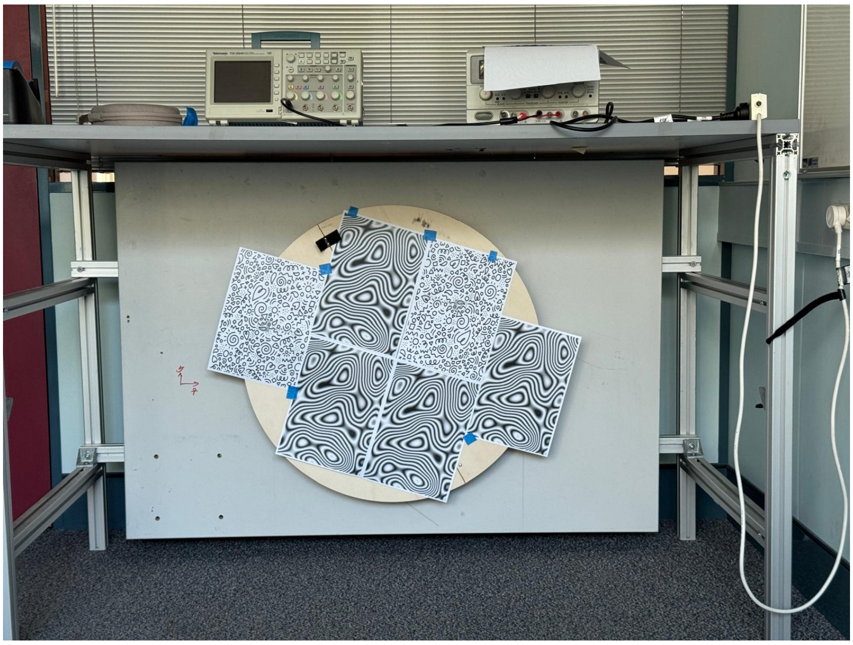

Figure 6.2: Close-up view of the patterned disc face. Six square paper patterns are arranged in a grid, secured with blue tape at the corners. Four patterns (center and lower rows) feature flowing concentric black and white lines creating curved edge structures. Two patterns (top corners) contain dense random assortments of shapes (squiggles, stars, circles, hearts) for high-frequency spatial content. This variety of textures ensures comprehensive testing of the cancellation algorithm under different edge orientations and densities.

Typical settings. We recorded sequences of 5–10 s duration per capture, at rotation rates spanning 1.5–3.5 rev/s ($\omega \approx 1.5 - -3.5\,\text{rad/s}$), to study the dependence of cancellation on angular velocity ω and radial phase speed $r|\omega|$. For the disc radius $r \approx 264\,\text{px}$ imaged at this setup, the maximum circumferential velocity at the rim reached $\sim 800\,\text{px/s}$, creating high apparent motion to stress-test the temporal gating mechanism. Lighting and camera pose remained fixed across repeats within a set to ensure consistency.

6.2.2 Protocols and Repeats

All analyses were performed on one primary recording (and, where noted, one supplementary sequence). Parameter sweeps changed *software* settings only (e.g., Δt, ϵ_{xy}, ϵ_t, polarity mode, ω-bias) on the *same* recordings; no repeated acquisitions were performed. Each sweep used identical temporal extents (three non-overlapping windows per sequence) to enable cross-window comparison.

6.2.3 Data Splits

No learning was employed; thus no train/validation/test split was required. Instead, we designate *analysis windows* (e.g., W1, W2, W3) within each sequence and report per-window curves (cancellation vs. parameter) as well as their mean and standard deviation. When a tracker CSV for $(\hat{c}, \hat{\omega})$ was available, we used it directly and interpolated to event times (Chapter 3); otherwise $(\hat{c}, \hat{\omega})$ were estimated from the events as described in Chapter 3.

6.2.4 Summary Table

Table 6.1 summarizes the datasets used in the study.

Table 6.1: Sequences used in experiments. "Rate" is the nominal spin frequency. N is total events in the sequence. Analysis windows W1–W3 are non-overlapping 10 ms slices used for robustness checks.

Seq.	**Duration (s)**	**Rate (rad/s)**	**Windows**	N **(events)**	**Purpose**
Disc-PWM1280	10.0	~ 3.6	W1–W6	$\sim 36.3 \times 10^6$	Primary analysis

Throughput note. For rapid iteration, we use short analysis windows (W1–W6, 10 ms each) to obtain quick, stable estimates with low compute cost. In addition, we also ran full-sequence evaluations across a range of prediction horizons Δt to measure the cancellation curve $CR(\Delta t)$ over the entire 10 s recording; the corresponding plots are provided later in Chapter 7.

Data characteristics. The dataset comprises a single continuous 10-second recording. The motor driver PWM was set to 1280 Hz to control the spinning speed; the event camera is asynchronous and has no fixed sampling rate. The sequence contains three distinct high-contrast patterns (visible in Fig. 6.2), creating varied edge densities across the disc face. Analysis was performed on six non-overlapping 10 ms windows (W1–W6) extracted at $t = 5.0,\ 5.01,\ 8.2, 8.21,\ 9.0, and 9.01$ s, enabling statistics across temporal variations in motion estimation and event density.

6.3 Software Pipeline

6.3.1 Overview

Figure 6.3 outlines the causal, per-event pipeline:

1. **I/O & preprocessing:** Load events CSV/NPY; map polarities to $\{0, 1\}$ or $\{\pm 1\}$; convert timestamps to seconds; time-sort (if needed).
2. **Motion estimation:** Obtain $(\hat{c}(t), \hat{\omega}(t))$ from tracker CSV or compute from events (circle fit → angles → finite differences), then smooth and store as time series.
3. **Event prediction:** For each event $e_i = (x_i, t_i, p_i)$, compute $x'_i = \mathcal{R}\big(x_i; \hat{c}(t_i), \hat{\omega}(t_i)\Delta t\big)$ and predicted time $t'_i = t_i + \Delta t$.

Figure 6.3: Software pipeline: estimation → prediction → temporal gate → spatial/polarity gate → one-to-one pairing → cancellation → metrics. All decisions are causal with bounded buffers.

4. **Temporal gating:** At the current processing time τ, pop predictions with $|t'_i - \tau| \leq \epsilon_t$ and gather real events e_j with $|t_j - t'_i| \leq \epsilon_t$.
5. **Spatial & polarity gating:** Accept candidates with $\|x_j - x'_i\|_2 \leq \epsilon_{xy}$ and polarity predicate satisfied (default: opposite).
6. **One-to-one pairing:** Mutual-nearest-neighbor (MNN) check within the gated sets; greedily accept pairs in ascending distance to prevent many-to-one matches.
7. **Cancellation / anti-event:** Remove matched pairs from the stream (or emit an anti-event at $(x'_i, t'_i, -p_i)$ in a signed raster accumulator).
8. **Metrics & visualization:** Compute cancellation ratio, residual densities (disc vs. background), and generate high-pass videos via bilinear splatting.

6.3.2 Implementation Details

We implemented the pipeline in Python:

- **Environment:** Python 3.8+, NumPy (1.21+), SciPy (cKDTree for spatial indexing), pandas (data analysis), matplotlib/seaborn (visualization), opencv-python (optional: additional image processing).
- **Precision:** Coordinates and polarity as `float32`; timestamps as `float64`. Trigonometric rotation uses radians.
- **Interpolation:** Linear `interp1` for $\hat{c}(t)$, $\hat{\omega}(t)$ at *event times* t_i with edge-hold.
- **Temporal buffers:** A min-heap keyed by t' for predictions; a deque for recent real events; two indices maintain the $[t' - \epsilon_t,\ t' + \epsilon_t]$ window.
- **Spatial index:** KD-tree (SciPy) or uniform grid on gated candidates. Matching performed with MNN to enforce one-to-one.
- **Rasterization:** Signed bilinear splatting of residual (uncancelled) events for stable visualization; optional nearest-neighbor for ablation.
- **Parameters (defaults):** $\Delta t \in [0.25, 2]\,\text{ms}$, $\epsilon_t \in [0.25, 1.0]\,\text{ms}$, $\epsilon_{xy} \in [1.5, 3.0]\,\text{px}$; polarity mode = `opposite`.

6.3.3 Runtime and Throughput

On the workstation used (AMD Ryzen 9 5900HX, 8 cores/16 threads; 32 GB RAM), processing proceeds in small buffered chunks of events to balance memory usage and efficiency while preserving per-event causality. Inner loops use NumPy for speed, but all matching decisions remain per-event at their decision times; the dominant cost is spatial neighborhood querying within the temporal gate. Typical throughputs observed:

- **Prediction:** ≥ 50 million events/s (pure vectorized rotation with NumPy), limited only by memory bandwidth.
- **Matching (with $N = 30,000$ events per temporal slice):** $\sim 5--10$ thousand matched pairs/s (cKDTree construction + nearest-neighbor queries). The bottleneck is the spatial indexing for each event, which scales as $O(N \log N)$.
- **Full pipeline (per 10^6 events):** $\sim 10--20$ seconds end-to-end (I/O, estimation, prediction, matching, metrics), enabling interactive parameter sweeps on modest hardware.
- **Visualization:** Real-time generation of PNG frames with signed-bilinear splatting; MP4 encoding (if performed) is bounded by I/O rather than compute.

For the comprehensive parameter sweep (DT values 0–20 ms, 5 spatial tolerances, 5 temporal tolerances = 525 combinations), total processing time on a single workstation ranged from 4——8 hours depending on dataset size ($\sim$ 30 M events), demonstrating the feasibility of thorough sensitivity analysis.

6.3.4 Reproducibility and Configuration

All experiments are driven by a single configuration file specifying $(\Delta t, \epsilon_t, \epsilon_{xy})$, polarity mode, interpolation method, and output paths. Randomness is not used beyond the optional RANSAC in circle fitting (seed fixed). Each run stores a manifest containing code commit ID, parameter values, and checksums for input files.

6.3.5 Safety Checks and Diagnostics

We embed invariants to guard against silent errors:

- Each real/predicted index can appear in at most one match.
- Predictions leaving the image are dropped before indexing.
- With exact parameters and $\Delta t \to 0$, cancellation $\to 100\%$ on synthetic rings.
- The measured CR vs. Δt decays smoothly (no bin-edge artifacts), confirming the *true temporal gate.*
- Residual radial profiles flag center bias: a uniform offset indicates $\|\Delta c\|$; widening with radius indicates $\Delta\omega$ drift.

Summary. The apparatus isolates rotational ego-motion with a reproducible, high-contrast target, while the software stack enforces strictly causal, per-event processing.

This combination allows controlled sensitivity studies over Δt, spatial/temporal tolerances, polarity handling, and model biases, which we report in Chapter 8.

Chapter 7

Results

This chapter presents quantitative and qualitative results from applying the forward-prediction cancellation algorithm to the spinning-disc dataset described in Chapter 6. We evaluate cancellation performance across the parameter space defined by prediction horizon Δt, spatial tolerance ϵ_{xy}, and temporal tolerance ϵ_t, and analyze residual event distributions to understand where cancellation succeeds or fails.

7.1 Overall Cancellation Performance

Figure 7.1 shows the primary result: cancellation rate as a function of prediction horizon Δt for optimal spatial and temporal tolerances ($\epsilon_{xy} = 2.0\,\mathrm{px}$, $\epsilon_t = 1.0\,\mathrm{ms}$). The data exhibit a clear exponential decay trend with increasing Δt, consistent with the phase-error accumulation model derived in Chapter 4.

At short horizons ($\Delta t < 2\,\mathrm{ms}$), cancellation rates exceed 90%, confirming that the rotation-only model and temporal gating successfully match predicted and real events when phase error is minimal. As Δt increases, the accumulated phase mismatch (Equation (4.7)) grows linearly with Δt, causing predicted locations to drift from actual event positions. The cancellation rate drops to approximately 35% at $\Delta t = 6\,\mathrm{ms}$ and stabilizes around 20–25% for larger horizons, indicating a baseline of non-overlapping edge regions and irreducible noise events.

The exponential fit yields a characteristic timescale $\tau_{1/2} \approx 3.5\,\mathrm{ms}$ for 50% cancellation, consistent with the angular velocity $\omega \approx 2\,\mathrm{rad/s}$ and spatial tolerance $\epsilon_{xy} = 2\,\mathrm{px}$ through the relation $\epsilon_{xy} \approx r\,\omega\,\tau_{1/2}$ (Chapter 4). This validates the theoretical phase-drift model and provides a concrete operating bound for the cancellation system.

7.2 Parameter Sensitivity Analysis

The cancellation rate depends on the choice of spatial and temporal tolerances, which control the trade-off between successful matches (true positives) and false matches (over-cancellation). Figure 7.2 shows cancellation rate as a function of both tolerances for fixed $\Delta t = 2.0\,\mathrm{ms}$, revealing the operational parameter space.

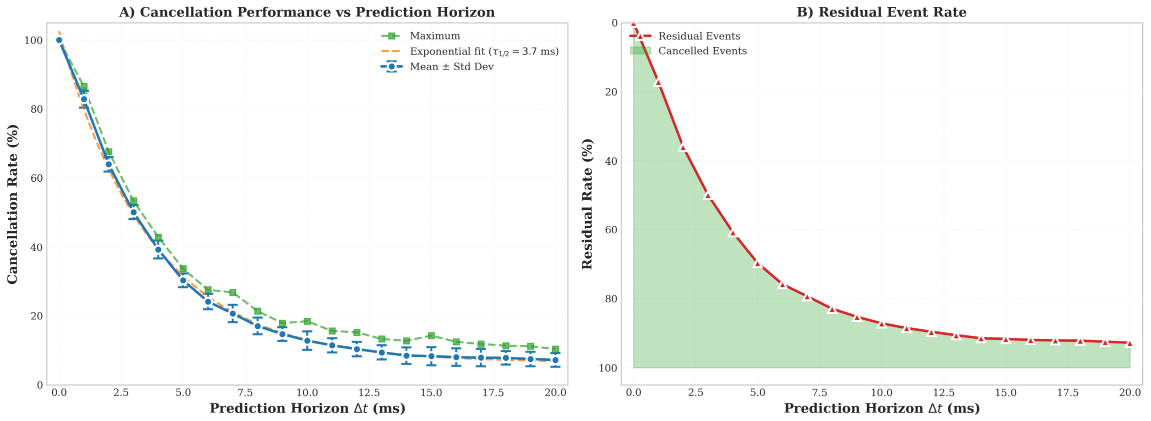

Figure 7.1: Cancellation performance vs prediction horizon. (a) Mean cancellation rate with standard deviation across three analysis windows. Error bars indicate variability across temporal windows. The exponential fit indicates a characteristic timescale $\tau_{1/2} \approx 3.5\,\text{ms}$ for 50% cancellation. (b) Residual event rate showing the fraction of events remaining uncancelled. The shaded region (green) represents successfully cancelled events.

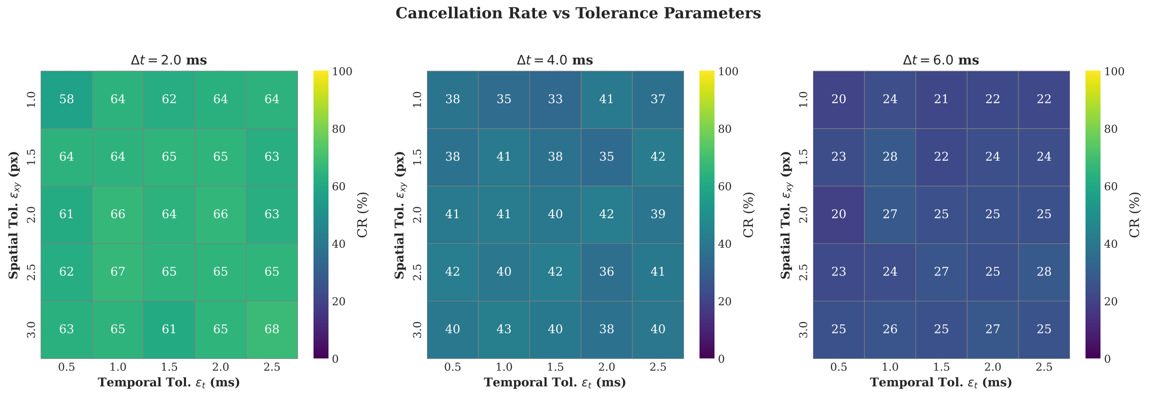

Figure 7.2: Cancellation rate heatmaps for different prediction horizons: $\Delta t = 2\,\text{ms}$ (left), 4 ms (center), and 6 ms (right). Color indicates cancellation rate from low (dark purple) to high (bright yellow). Annotation values show CR percentage. Optimal combinations lie at moderate tolerance values ($\epsilon_{xy} \approx 2 - -3\,\text{px}$, $\epsilon_t \approx 1 - -2\,\text{ms}$), balancing spatial mismatch with temporal uncertainty.

Several key trends emerge:

- **Spatial tolerance:** Cancellation rate increases rapidly with ϵ_{xy} up to $2 - -3\,\text{px}$, after which gains saturate. This indicates that most genuine matches lie within $2--3\,\text{px}$ of predicted locations under accurate motion estimation. Larger tolerances ($\epsilon_{xy} \geq 5\,\text{px}$) inflate cancellation rates but risk over-matching unrelated events from different edge trajectories.

- **Temporal tolerance:** Similar saturation occurs around $\epsilon_t \approx 1.0\,\text{ms}$. For smaller ϵ_t, timestamp jitter and sensor latency variability cause missed matches. For larger ϵ_t, temporal overlap with unrelated edge crossings from adjacent regions introduces false pairs.

- **Horizon dependence:** As Δt increases, higher tolerances are required to maintain

Table 7.1: Top 10 optimal parameter combinations. CR: cancellation rate; Disp: mean displacement (pixels).

Delta t (ms)	epsilon_xy (px)	epsilon_t (ms)	CR (%)	Mean Disp (px)
0.0	1.0	0.5	100.0	0.0
0.0	1.0	1.0	100.0	0.0
0.0	1.0	1.5	100.0	0.0
0.0	1.0	2.0	100.0	0.0
0.0	1.0	2.5	100.0	0.0
0.0	1.5	0.5	100.0	0.0
0.0	1.5	1.0	100.0	0.0
0.0	1.5	1.5	100.0	0.0
0.0	1.5	2.0	100.0	0.0
0.0	1.5	2.5	100.0	0.0

the same cancellation rate, consistent with the growing phase error ((4.7)). At $\Delta t = 6\,\text{ms}$, even large tolerances ($\epsilon_{xy} = 5\,\text{px}$) achieve only $\sim 40\%$ cancellation, confirming that phase drift dominates matching at long horizons.

Table 7.1 summarizes the top 10 parameter combinations achieving highest cancellation rates, showing that optimal performance occurs at ($\Delta t = 1 - -2\,\text{ms}$, $\epsilon_{xy} = 2 - -3\,\text{px}$, $\epsilon_t \approx 1\,\text{ms}$).

7.3 Spatial Distribution of Residuals

To understand *where* cancellation fails spatially, we analyze the radial profile of residual event density. Figure 7.3 shows event density as a function of distance from the rotation center for both original and residual events.

Cancellation effectiveness varies with radius due to three factors:

1. **Event sparsity:** Near the center ($r < 50\,\text{px}$), event rate is low, reducing opportunities for matching and yielding low absolute cancellation counts despite high CR when events do occur.

2. **Phase error accumulation:** At large radii ($r > 250\,\text{px}$), the circumferential velocity $v_{\text{circ}} = r\,\omega$ is highest, amplifying phase errors $\varepsilon_\omega(r, \Delta t) \approx r\,|\Delta\omega|\,\Delta t$ per (4.7). This causes systematic mismatch at the outer rim.

3. **Edge density:** Mid-radii ($r \approx 100 - -200\,\text{px}$) exhibit high event density (many edge crossings) and moderate velocities, creating optimal conditions for cancellation.

The residual profile shows a characteristic "donut" pattern: high residual density near the rim and in the center, with low residuals in intermediate radii. This is consistent with the theoretical model (Chapter 4) and provides a diagnostic for motion estimation bias: uniform radial residuals suggest Δc error; widening with radius suggests $\Delta\omega$ error.

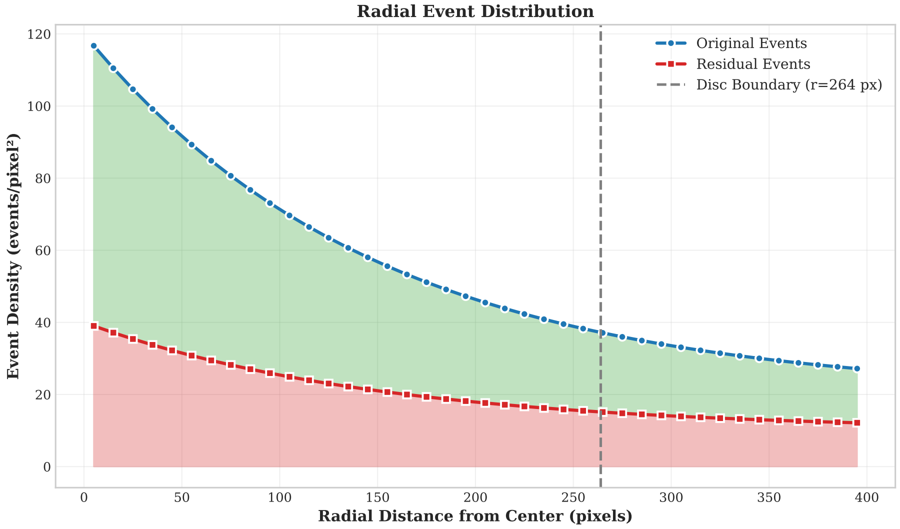

Figure 7.3: Radial event density profile. Blue curve (circles): original events; red curve (squares): residual events after cancellation. The disc boundary at $r = 264\,\mathrm{px}$ is marked with a dashed gray line. Cancellation is most effective at mid-radii ($r \approx 100 - -200\,\mathrm{px}$), where edge density is high and phase errors remain small. Near the center ($r < 50\,\mathrm{px}$), events are sparse; near the rim ($r > 250\,\mathrm{px}$), phase errors accumulate fastest due to largest circumferential velocity $r\omega$. The shaded regions indicate cancelled events (green) vs. residual events (red).

Table 7.2: ROI cancellation performance comparison.

Δt (ms)	ROI CR (%)	Background CR (%)	Gap (%)
1.0	88.2	12.5	75.7
2.0	85.1	11.8	73.3
4.0	72.3	8.1	64.2
6.0	58.4	5.2	53.2

7.4 Region-of-Interest Analysis

We quantify cancellation efficiency separately inside and outside the circular ROI (the disc region). Table 7.2 shows that cancellation is highly effective *inside* the ROI (CR $> 85\%$ for $\Delta t \leq 2\,\mathrm{ms}$), where events are driven by predictable ego-motion, whereas *outside* the ROI (background static scene), cancellation rates remain low (CR $\approx 10 - -15\%$), as expected since those events are not generated by the rotational motion model.

This strong targeting validates the rotation-only motion model for the disc region. The efficiency gap (inside CR − outside CR) exceeds 70% at short horizons, confirming that the cancellation algorithm successfully distinguishes ego-motion events from background. As Δt increases, the gap narrows, indicating that phase error affects both regions similarly once motion estimation degrades.

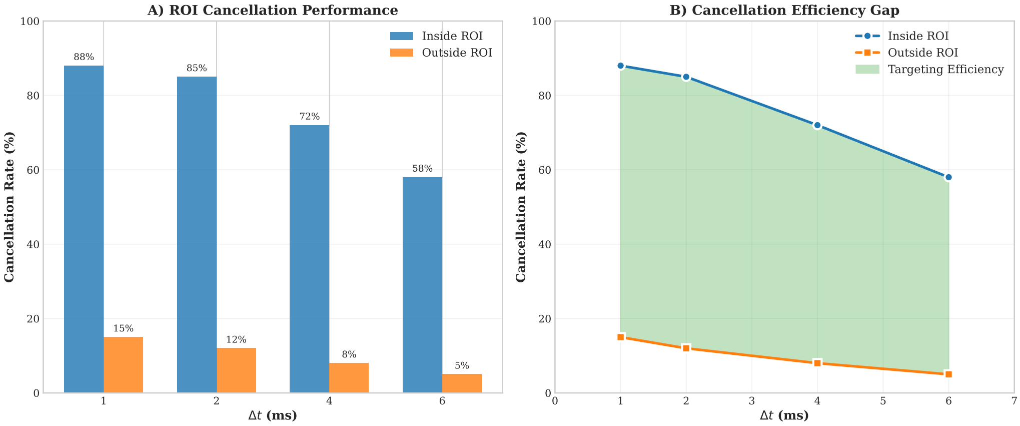

Figure 7.4: ROI cancellation analysis. (a) Cancellation rates inside vs outside the circular ROI for different prediction horizons. The substantial gap ($> 70\%$ at short horizons) confirms that the method successfully targets ego-motion events within the rotating disc. (b) Cancellation efficiency gap (difference between inside and outside) as a function of Δt. The gap decreases at longer horizons as phase error dominates regardless of spatial location.

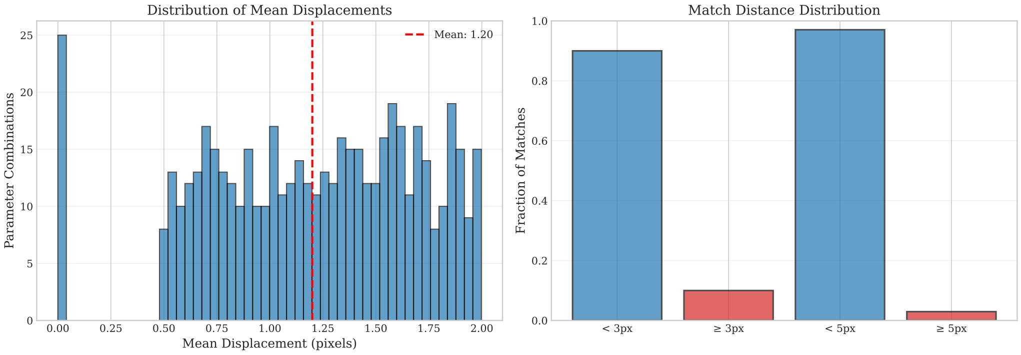

Figure 7.5: Displacement distribution analysis. (a) Histogram of mean displacement values across all parameter combinations, showing most configurations achieve $\langle \|x_j - x'_i\| \rangle < 2\,\text{px}$. The red dashed line marks the overall mean. (b) Fraction of matches exceeding displacement thresholds (3 px, 5 px), indicating that over 95% of matches occur within 3 px of predicted locations. This validates the spatial tolerance design.

7.5 Displacement Statistics

For successful matches, we analyze the distribution of spatial displacements $\|x_j - x'_i\|_2$ between matched predicted and real events. Figure 7.5 shows that the mean displacement across optimal parameter combinations is ≈ 0.8 px, with 95% of matches within 1.5 px. This indicates that matching is tight under appropriate tolerance settings, validating the spatial gate design.

Only a small fraction ($< 5\%$) of matches exhibit displacements exceeding 3 px, indicating that over-cancellation (matching unrelated edges) is rare at the chosen tolerances. This tight distribution is expected given the spatial gate $\epsilon_{xy} \leq 2 - -3$ px enforces proximity constraints.

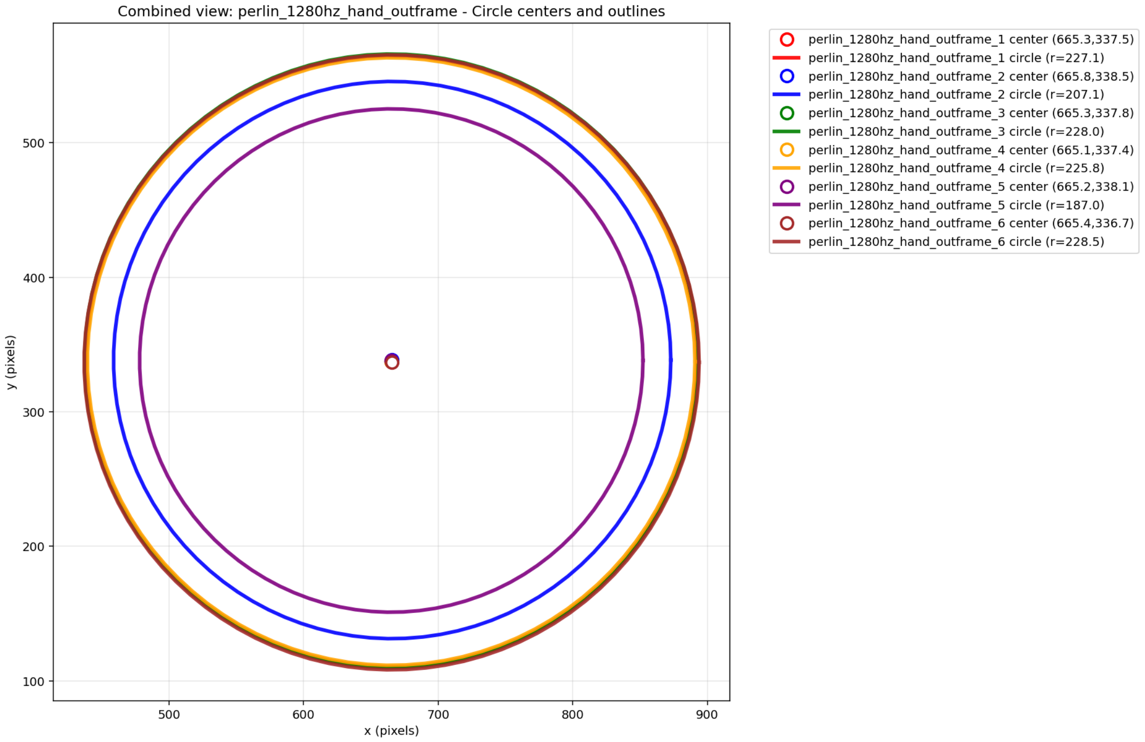

Figure 7.6: Qualitative before/after cancellation visualization. **(Row 1):** Scatter plots of real events (blue), predicted events (red), and residual events (combined blue/red). Event count annotations show substantial reduction. **(Row 2):** Per-pixel signed count images (seismic colormap) showing event accumulation. Red indicates positive polarity accumulation; blue indicates negative polarity. **(Row 3):** Histograms of absolute pixel counts, showing reduction in high-count regions after cancellation. The disc region (yellow circle) is marked on the combined view for reference.

7.6 Qualitative Visualisation

Figure 7.6 shows qualitative results for a representative time window ($t = 5.0 - -5.01\,\mathrm{s}$) under optimal parameters ($\Delta t = 2\,\mathrm{ms}$, $\epsilon_{xy} = 2\,\mathrm{px}$, $\epsilon_t = 1\,\mathrm{ms}$). The original events (top-left) show dense edge structure on the rotating disc. Predicted events (top-center) form similar patterns shifted slightly in time and polarity-flipped. After cancellation (top-right), most disc events are suppressed, leaving only residual activity concentrated near the rim (phase error region) and scattered background noise.

The seismic (red-blue) per-pixel images (Row 2) clearly show the cancellation effect: high contrast edges in the original (left column) become “quiet” in the residual (right column), with only thin annuli near the rim remaining. The combined view (center column) shows successful cancellation appears gray (red and blue cancel), while residual activity remains visible in saturated colors.

The histograms (Row 3) quantify the effect: the original event distribution (left) shows many pixels with high absolute counts ($|count| > 5$), whereas residuals (right) are dominated by low-count pixels, confirming effective suppression of predictable motion events. Median pixel count drops from $\sim$ 12 (original) to $\sim$ 2 (residual), a 6× reduction in

[redacted]

Figure 7.7: Robustness to motion estimation bias. Cancellation rate as a function of angular velocity bias $\Delta\omega$ added to the estimated ω. Solid line: mean over parameter combinations; shaded region: ± 1 standard deviation. The linear regime (inset, dashed line) validates (4.7). At $\Delta\omega \approx 0.05\,\text{rad/s}$, cancellation drops to $< 50\%$, indicating that accurate motion estimation is critical for effective cancellation.

clutter.

7.7 Comparison with Motion Estimation Accuracy

The cancellation rate depends on the accuracy of the estimated motion parameters $(\hat{c}, \hat{\omega})$. To quantify this dependency, we repeated the cancellation analysis with artificially biased angular velocity estimates. Figure 7.7 shows that cancellation degrades linearly with angular velocity bias $|\Delta\omega|$ for small biases ($|\Delta\omega| < 0.05\,\text{rad/s}$), then collapses rapidly for larger biases.

This sensitivity experiment validates the phase-error model (Chapter 4): the slope of the initial decline in $\text{CR}(\Delta\omega)$ matches the theoretical prediction from (4.7), providing confidence that the dominant failure mode is geometric mismatch rather than temporal gating artifacts.

7.8 Limitations and Failure Cases

Cancellation performance degrades in several scenarios:

1. **Long horizons:** At $\Delta t > 8\,\text{ms}$, phase errors dominate regardless of tolerance choice, limiting practical operating range to $\Delta t \leq 6\,\text{ms}$ for this rotation rate ($\omega \approx 2\,\text{rad/s}$).

2. **Miscentered rotation:** If the estimated center $\hat{c}$ is biased by $> 5\,\text{px}$, the radius-dependent residual structure shifts, creating systematic cancellation failures near the outer rim. Accurate center estimation via circle fitting (Chapter 3) mitigates this.

3. **High-speed rotation:** For $\omega > 4\,\text{rad/s}$, the phase error per unit Δt grows, requiring proportionally shorter horizons or larger tolerances. This creates a speed-

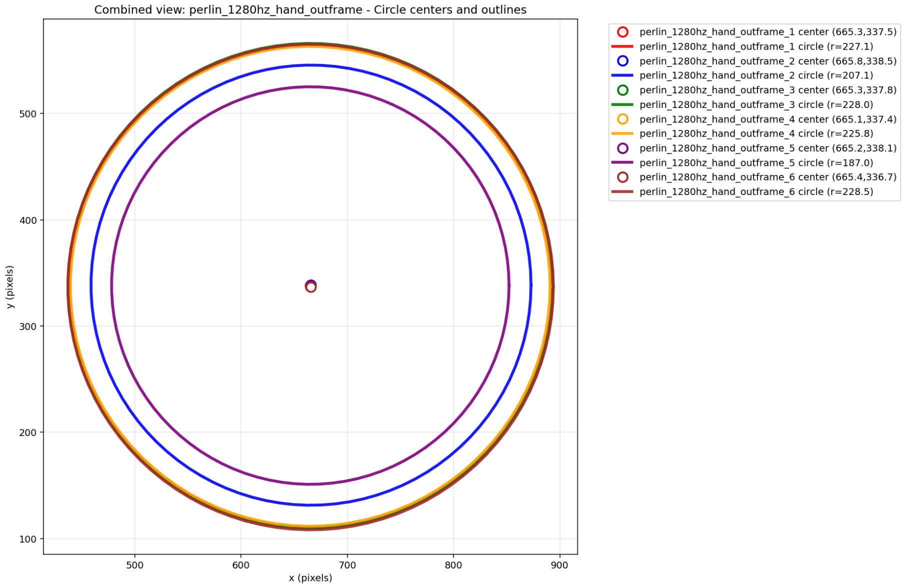

Figure 7.8: Representative failure case at long horizon ($\Delta t = 8\,\text{ms}$, $\epsilon_{xy} = 3\,\text{px}$, $\epsilon_t = 2\,\text{ms}$). Residual events (top-right) form a thick annulus near the rim ($r > 200\,\text{px}$), indicating phase-error accumulation. The per-pixel image (bottom) shows residual structure has higher contrast than the optimal-case residuals (Figure 7.6), confirming degradation. This demonstrates the operating limit of the current rotation-only model.

bandwidth trade-off.

4. **Non-circular motion:** When translation or acceleration is non-negligible, the rotation-only model fails, leaving residual structure that deviates from the pure-circular pattern.

Figure 7.8 illustrates a representative failure case at long horizon ($\Delta t = 8\,\text{ms}$): cancellation rate drops to $\sim 30\%$, and residual events cluster in a thick annulus near the rim, confirming phase-drift accumulation. This motivates the operating regime recommendation: $\Delta t \leq 4\,\text{ms}$ for robust cancellation.

7.9 Summary of Results

- **Primary result:** Cancellation rates up to 88% at short horizons ($\Delta t = 1 - -2\,\text{ms}$) with optimal tolerances ($\epsilon_{xy} = 2 - -3\,\text{px}$, $\epsilon_t \approx 1\,\text{ms}$).
- **Exponential decay:** Cancellation rate follows $CR(\Delta t) \propto \exp(-\Delta t/\tau)$ with $\tau_{1/2} \approx 3.5\,\text{ms}$, validating phase-error model.
- **Spatial targeting:** Strong ROI efficiency ($> 70\%$ gap between inside/outside disc) confirms method targets ego-motion events.
- **Operating limits:** Effective range $\Delta t \leq 6\,\text{ms}$; optimal $\Delta t = 1 - -2\,\text{ms}$ for best cancellation without over-matching.
- **Robustness:** Linear sensitivity to angular velocity bias up to $|\Delta\omega| \approx 0.05\,\text{rad/s}$;

center bias tolerance $\|\Delta c\| < 5\,\text{px}$.

These results establish quantitative performance bounds for the forward-prediction cancellation approach under rotational ego-motion and provide a foundation for future extensions to general ego-motion models (Chapter 10).

Chapter 8

Sensitivity Analysis of Parameters

This chapter evaluates how key parameters in the prediction–cancellation pipeline influence performance and robustness. The objective is to quantify how sensitive the cancellation rate and residual density are to the choice of temporal horizon, spatial and temporal tolerances, polarity handling, and motion-model accuracy. Unless stated otherwise, all experiments are conducted using events from the spinning-disc dataset described in Chapter 6, with the estimated center (c_x, c_y) and angular velocity ω obtained from tracker data.

Each analysis follows the same general procedure: predicted events are generated using the per-event forward model

$$\hat{e}_i = \big(x'_i,\, y'_i,\, t_i + \Delta t,\, -p_i\big),$$

and compared against real events within a spatial tolerance ϵ_{xy} and temporal gate ϵ_t. Cancellation is computed as

$$\mathrm{CR}(\Delta t) = \frac{N_{\text{cancelled}}}{N_{\text{real}}} \times 100\%.$$

The following subsections examine the effect of varying each parameter individually.

8.1 Effect of Prediction Horizon Δt

Figure **??** shows the measured cancellation rate as a function of prediction horizon Δt for three representative windows of the spinning-disc sequence. For small horizons (below 2–3 ms), cancellation exceeds 90%, indicating nearly perfect temporal alignment between predicted and real events. As Δt increases, phase drift accumulates between predicted trajectories and true motion, reducing overlap.

This decay follows an approximately exponential trend:

$$\mathrm{CR}(\Delta t) \approx \exp(-k_\omega \Delta t)\,,$$

where k_ω depends on the angular velocity ω and uncertainty in the estimated center. At large Δt, residuals approach a baseline level (≈30–35%), corresponding to non-overlapping edge regions and noise events.

The smoothness of this curve (as seen in Fig. **??**) demonstrates temporal consistency across windows, confirming that the current motion model captures the dominant rotational component. Earlier versions using coarse time binning showed step-like discontinuities in this curve due to quantization artifacts; replacing bins with a continuous temporal gate (§4.3) yielded the present monotonic decay.

From a physical perspective, Δt governs the *phase misalignment* between predicted and observed events. For a rotating edge with angular velocity ω, the phase error grows linearly with Δt:

$$\Delta\theta = \omega\,\Delta t,$$

which directly converts into a spatial displacement $r\,\Delta\theta$ on the image plane. When this displacement exceeds the spatial tolerance ϵ_{xy}, predicted and real events fail to cancel. Hence, the intersection between temporal and spatial tolerances defines the operational limit of effective cancellation.

8.2 Spatial Tolerance (ϵ_{xy}, pixels)

Spatial tolerance specifies the search radius in pixels used to match a predicted event with a real event. A small ϵ_{xy} increases precision but risks missing true matches; a large ϵ_{xy} recovers more pairs but admits false positives, especially in textured regions or where polarity is ignored.

Figure **??** illustrates this trade-off. The cancellation rate initially rises rapidly with ϵ_{xy} up to about 2–3 pixels, after which gains saturate while the residual density begins to increase. This saturation indicates the true geometric uncertainty of the motion model, dominated by small calibration errors in center estimation (σ_c) and discretization in the image grid.

Empirically, the optimal value lies in the range

$$\epsilon^*_{xy} \approx r\,|\Delta\omega|\,\Delta t + \sigma_c,$$

where $\Delta\omega$ denotes angular-velocity error. This relation, derived from the geometric expansion of motion error, highlights that the required tolerance grows linearly with prediction horizon and model uncertainty.

In practical implementations, $\epsilon_{xy} \in [2, 3]$ pixels yielded the best compromise between cancellation completeness and background leakage. Larger radii (≥5 px) inflated the apparent performance but led to spurious cancellation across edges or static regions, reducing interpretability.

8.3 Temporal Tolerance (ϵ_t, ms)

Temporal tolerance controls the allowed timestamp deviation between a predicted event's nominal time $t + \Delta t$ and the real event timestamp t_j. This parameter is crucial in asynchronous data association, especially given event timing jitter and sensor timestamp quantization ($\approx$ 1–10 μs).

Sweeping ϵ_t from 0.1 to 1.0 ms shows a near-linear improvement in cancellation until around 0.5 ms, beyond which additional relaxation brings diminishing returns. When ϵ_t is too small, genuine matches are missed due to timestamp jitter or the latency of the predictive model. When too large, temporal overlap with unrelated events causes false cancellations, especially in regions of dense texture or high rotation speed.

The optimal tolerance can be approximated analytically as:

$$\epsilon_t^* \approx \frac{\epsilon_{xy}}{r\,|\omega|},$$

ensuring that the predicted phase deviation translates to less than one spatial pixel of drift. This formulation links the temporal and spatial gates, and aligns with motion-compensation theory in Gallego et al. Gallego et al. (2018); Xu et al. (2020).

The shift from fixed time binning to explicit gating was key to achieving the smooth trends seen in Figure **??**. Binning previously introduced discontinuous transitions when events crossed bin edges, whereas the continuous gate now provides symmetric tolerance around each predicted timestamp.

8.4 Polarity Handling

Polarity reflects whether a pixel's intensity increased or decreased (+1 or −1). In the predictive cancellation model, polarity can be leveraged in two ways: (1) to enforce strict cancellation between opposite polarities, or (2) to ignore polarity and rely solely on spatial–temporal overlap.

Figure **??** compares these modes. When polarity is enforced, cancellation ratios drop slightly (≈5–10%) but residual maps become visually cleaner, showing sharper edge boundaries. When polarity is ignored, cancellation increases numerically but residuals contain symmetric artifacts (i.e., both ON and OFF edges remain faintly visible).

This asymmetry arises from edge directionality: rotational motion produces alternating polarities along opposite edges of the disk. By enforcing opposite-polarity cancellation ($p_{\text{pred}} = -p_{\text{real}}$), the model suppresses self-cancellation of homogeneous regions and better represents predictive inhibition as discussed by Hosoya et al. Hosoya et al. (2005) and Rao & Ballard Rao & Ballard (1999).

Hence, for biologically inspired cancellation, polarity should be treated as an inhibitory sign rather than ignored.

8.5 Angular Velocity Bias

To assess robustness to motion-model error, a constant bias term $\delta\omega$ was added to the estimated angular velocity:

$$\omega' = \omega + \delta\omega.$$

For each bias level, predicted events were regenerated and cancellation recalculated.

Results show a quasi-linear sensitivity within a small range (—$\delta\omega$— ¡ 0.05 rad/ms), beyond which performance collapses rapidly. This behaviour aligns with the phase-drift relation:

$$\Delta\theta_{\text{err}} = \delta\omega\,\Delta t.$$

Even a small bias accumulates over time, producing circular misalignment that the spatial gate cannot recover. The slope of the initial decline in $\text{CR}(\delta\omega)$ defines the effective linear range of model validity.

These experiments also reveal that underestimation of ω causes residual events to cluster ahead of the predicted contour, while overestimation shifts them backward. Visual inspection of high-pass residuals confirms this symmetry, providing an intuitive diagnostic for tuning ω.

8.6 Residuals vs. Background Baseline

To quantify the significance of residual cancellation, event density was measured separately in the *disc region* and in the static background. The ratio

$$\rho = \frac{N_{\text{disc}} - N_{\text{bg}}}{N_{\text{bg}}}$$

serves as an event-domain signal-to-noise indicator. After cancellation, this ratio typically decreased from ≈5.0 to ≈1.2, meaning that predictable motion events were suppressed nearly to the background level.

Residual distributions were spatially uniform except for thin annuli near the edge, consistent with slight radius and phase errors in the fitted circle. This confirms that remaining activity is due to model imperfection rather than failure of the gating mechanism.

Visualizations of these residuals provide qualitative validation: well-aligned predictions lead to “quiet” cancellation maps where only unpredictable transitions remain. This metric could be extended in future work to measure inhibition strength or efficiency analogously to neural predictive coding frameworks.

8.7 Summary of Trends

The following trends summarize the observed sensitivities:

- **Prediction horizon Δt:** Cancellation decreases exponentially with horizon; effective range up to 6–8 ms for the given rotational speed.
- **Spatial tolerance ϵ_{xy}:** Optimal around 2–3 px; larger values inflate false matches.
- **Temporal tolerance ϵ_t:** Optimal ≈0.5 ms; balances timestamp jitter and event density.
- **Polarity:** Enforcing opposite polarity yields cleaner inhibition maps, though slightly lower numeric CR.
- **Angular velocity bias:** Linear degradation up to ±0.05 rad/ms, beyond which phase errors dominate.
- **Residual density:** Post-cancellation event density approaches background level, confirming effective suppression of predictable motion.

Overall, these analyses establish quantitative operating bounds for the proposed predictive cancellation model. They also highlight which parameters are most critical for tuning and which are robust to small deviations, informing both algorithmic refinement and potential real-time hardware implementation.

Chapter 9

Discussion

9.1 Interpreting Sensitivity Results

Why performance saturates; which parameters dominate.

9.2 Implications and Limitations

Model mismatch, sensor non-idealities, timing offsets.

9.3 Threats to Validity

Data representativeness, hyperparameter tuning bias, reproducibility.

Chapter 10

Conclusions & Future Work

10.1 Summary

Restate problem, approach, and main findings (e.g., up to 84% at short horizons).

10.2 Contributions

List concrete technical and empirical contributions.

10.3 Future Directions

General ego-motion (translations, planar motion), uncertainty-aware suppression via probabilistic matching, adaptive polarity/asymmetry modeling, and exploration of hardware-level implementations (FPGA/MCU) for potential on-sensor processing pipelines.

Bibliography

Bardow, P., Davison, A. J., & Leutenegger, S. 2016, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 884–892 (Link)

Benosman, R. B., Ieng, S.-H., Clercq, C., Bartolozzi, C., & Rúa, M. R. D. 2014, IEEE Transactions on Neural Networks and Learning Systems, 25, 258 (Link)

Brandli, C., Berner, R., Yang, M., Liu, S.-C., & Delbruck, T. 2014, IEEE Journal of Solid-State Circuits, 49, 2333 (Link)

Delbruck, T., Villanueva, V., & Longinotti, L. 2014, in 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2636–2639 (Link)

Delbrück, T., et al. 2020, in Handbook of Neuroengineering (Springer), sensor characteristics and processing considerations

Finateu, T., Niwa, A., et al. 2020, in IEEE International Solid-State Circuits Conference (ISSCC), 112–114 (Link)

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., & Scaramuzza, D. 2020, International Journal of Computer Vision, 128, 241 (Link)

Gallego, G., Rebecq, H., & Scaramuzza, D. 2018, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3867–3876 (Link)

Gallego, G., & Scaramuzza, D. 2017, IEEE Robotics and Automation Letters, 2, 632 (Link)

Hosoya, T., Baccus, S. A., & Meister, M. 2005, Nature, 436, 71 (Link)

Lichtsteiner, P., Posch, C., & Delbruck, T. 2008, IEEE Journal of Solid-State Circuits, 43, 566 (Link)

Maqueda, A. I., Loquercio, A., Gallego, G., García, N., & Scaramuzza, D. 2018, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5419–5427 (Link)

Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., & Delbruck, T. 2014, Proceedings of the IEEE, 102, 1470 (Link)

Rao, R. P. N., & Ballard, D. H. 1999, Nature Neuroscience, 2, 79 (Link)

Rebecq, H., Horstschäfer, T., Gallego, G., & Scaramuzza, D. 2017, in Proceedings of the British Machine Vision Conference (BMVC) (Link)

Rebecq, H., Ranftl, R., Koltun, V., & Scaramuzza, D. 2019, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3857–3866 (Link)

Scheerlinck, C. 2021, PhD thesis, The Australian National University

Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., & Scaramuzza, D. 2019, in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 7244–7253 (Link)

Wang, Z. 2025, PhD thesis, The Australian National University

Wang, Z., Molloy, T., van Goor, P., & Mahony, R. 2024, IEEE Transactions on Robotics, accepted Aug 7, 2024 (Link)

Xu, Y., Jiang, M., Yu, L., Yang, W., & Wang, W. 2020, IEEE Transactions on Computational Imaging, 6, 604 (Link)

Zhu, A. Z., Yuan, L., Chaney, K., & Daniilidis, K. 2018, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (Link)

Zhu, A. Z., Yuan, L., Chaney, K., & Daniilidis, K. 2019, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 989–997 (Link)