



## GNEP BASED DYNAMIC SEGMENTATION AND MOTION ESTIMATION FOR NEUROMORPHIC IMAGING

HARBIR ANTIL<sup>✉\*</sup> AND DAVID SAYRE<sup>✉</sup>

The Center for Mathematics and Artificial Intelligence (CMAI) and  
 Department of Mathematical Sciences,  
 George Mason University, Fairfax, VA 22030, USA

**ABSTRACT.** This paper explores the application of event-based cameras in the domains of image segmentation and motion estimation. These cameras offer a groundbreaking technology by capturing visual information as a continuous stream of asynchronous events, departing from the conventional frame-based image acquisition. We introduce a Generalized Nash Equilibrium based framework that leverages the temporal and spatial information derived from the event stream to carry out segmentation and velocity estimation. To establish the theoretical foundations, we derive an existence criteria and propose a multi-level optimization method for calculating equilibrium. The efficacy of this approach is shown through a series of experiments.

**1. Introduction.** Event (Neuromorphic) cameras are a class of biologically inspired sensors that employ an asynchronous sampling mechanism to record data based on the change in light intensity at each pixel [1, 7, 21]. Specifically, an event is triggered at a pixel if and only if the change in light intensity at that pixel exceeds a predefined threshold. As a consequence, no events are recorded if there is no change to the scene. In contrast, if the scene is dynamic, either due to camera movement or due to the movement of an object in the scene each pixel of the event camera records intensity changes with a temporal resolution on the order of microseconds [7, 20]. Mathematically, an event camera can be modeled as a spatio-temporal point process, where the occurrence of an event at a pixel is a binary-valued function of time [20]. In [1], the authors introduced a mathematically rigorous bilevel optimization model, including a complete analysis, to carry out reconstructions from the event data. This model can use both a combination of traditional frames and event data. Figure 1 illustrates the differences between traditional and neuromorphic cameras.

Event image segmentation has emerged as a topic of interest, offering promising solutions to the challenging task of segmenting dynamic scenes captured by event-based cameras. Under the assumption of constant illumination, event occurrences

---

2020 *Mathematics Subject Classification.* Primary: 65K05, 90C26, 90C46; Secondary: 49J20.

*Key words and phrases.* Generalized Nash equilibrium, GNEPs, neuromorphic imaging, segmentation, velocity tracking, existence of solution.

This work is partially supported by NSF grant DMS-2110263, Air Force Office of Scientific Research (AFOSR) under Award NO: FA9550-22-1-0248, and Office of Naval Research (ONR) under Award NO: N00014-24-1-2147.

\*Corresponding author: Harbir Antil.

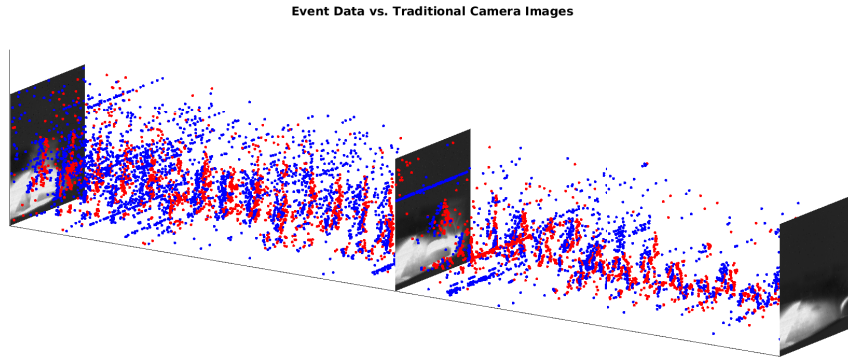


FIGURE 1. Comparison of the output of a standard frame-based camera (3 frames are shown) and event camera (dots indicate the events). The standard camera outputs frames at a fixed rate, thus sending redundant information when there is no motion in the scene. However, the event based camera captures the change in scene. For a model that can utilize all this information to carry out high quality reconstructions, we refer to [1].

in event-based cameras can be attributed to the relative motion between the camera and the observed scene [8, 22]. As events are primarily indicative of changes in brightness, they predominantly arise from scene edges, including contours, texture variations, and similar visual characteristics [25]. In the case of a stationary camera, events solely originate from moving objects within the scene [18]. However, when the camera itself undergoes motion, events are generated by both the edges of moving objects and the motion-induced edges of stationary objects [16]. This interplay between the camera's motion and the inherent dynamics of the scene adds complexity to event-based image segmentation, necessitating robust approaches to effectively distinguish between these sources of event occurrences. Various methods have been proposed to solve these problems including learning based [18, 27], probabilistic [26] and contrast maximization models [9, 10, 20, 21, 22, 23, 25].

Contrast maximization has emerged as a promising technique for event image segmentation. At its core, this approach focuses on finding a motion parameter such that the contrast is maximized when events are projected to an image plane. In the case of multiple objects a secondary task of discerning which events belong to which object must be undertaken [20, 25]. While these models/methods represent a leap forward but they are largely heuristic in nature and various questions remain about these models such as the existence of solutions, initialization of parameters, and optimization procedures to find solutions.

In this paper we provide alternative formulation to the previous methods by introducing a game theoretic model based on a Stackelberg Game. The presented approach is mathematically and algorithmically rigorous. A Stackelberg game is a type of non-cooperative game in game theory, named after the German economist Heinrich von Stackelberg. In a Stackelberg game, players make their decisions sequentially, with one player known as the leader or the Stackelberg player, acting first, and the other player or players, known as the followers, making their decisions after observing the leader's choice [12, 13, 15, 17], see also [4] for complexity guarantees in the context of multi-leader and multi-follower games. Stackelberg games

are often modeled mathematically as a Nash Equilibrium Problem or a Generalized Nash Equilibrium problem [3, 12, 24]. The Generalized Nash Equilibrium problem (GNEP) provides a mathematical framework for modeling a broad range of real-world scenarios involving strategic interactions among multiple autonomous agents. Specifically, GNEPs capture situations in which agents pursue individual objectives that are interdependent, thereby necessitating the consideration of others' decisions in determining their own optimal actions [2, 6].

In contrast to previous work we define the motion parameter as a function of the neuromorphic events versus defining the events as a function of their associated motion parameter. Our players will adopt a strategic approach by assigning a confidence value to each event within the dataset, indicating their level of certainty regarding the association between events. Using the confidence values for the events we then use linear regression to estimate the motion parameter that best fits the events. Furthermore, utilizing contrast maximization techniques we then evaluate the chosen strategy of each player. This allows us to identify the events that were generated by the same object and perform event based image segmentation. We enforce a restriction on the sum of confidence scores assigned to each event, limiting it to a maximum value of one. This results in a jointly constrained strategy space. In order to model this problem effectively we resort to the Generalized Nash Equilibrium model versus the Nash Equilibrium. In the remainder of the paper we seek to:

1. Introduce a Generalized Nash Equilibrium Problem for the segmentation of events.
2. Establish existence of equilibrium for the proposed model.
3. Propose and validate an  $N$ -level optimization method to find equilibrium.
4. Demonstrate the effectiveness of the model.

**Connection to SciML:** In recent years, the field of Scientific Machine Learning (SciML) has emerged as a transformative approach at the intersection of science, engineering, and machine learning. At its core, SciML embodies the fusion of scientific principles and data-driven modeling techniques to unravel complex phenomena. Scientific investigations often necessitate the extraction of valuable insights from vast datasets, and this is where SciML thrives. By harnessing machine learning algorithms and optimization methods, SciML empowers researchers to not only uncover hidden patterns within data but also to ground these discoveries firmly in the context of established scientific principles.

However, applying SciML to certain problems, such as Generalized Nash Equilibrium Problems (GNEPs) with players having non-convex objective functions, presents formidable challenges. The inherent non-convexity of these objectives precludes the application of traditional methods, such as quasi-variational inequality approaches, which rely on convexity both in the objective functions and the joint convexity of the strategy space. Consequently, alternative techniques become imperative to seek equilibrium solutions in such complex scenarios. One such approach is multi-level optimization, a strategy that dissects the problem into multiple levels and systematically optimizes them iteratively.

In this context, each optimization problem utilizes the SciML methodology by leveraging domain specific knowledge about neuromorphic imaging and the dynamics of event-based data. These dynamics can be thought as physics based model given by a discrete dynamical system (see [1, Eq. (2.6)]). Additionally, we employ a more traditional SciML technique (linear regression) to precisely estimate motion

parameters from the chosen data and perform image segmentation. This results in our ability to extract meaningful information from the complex neuromorphic data. Even though we are not using advanced SciML techniques (such as deep learning) here, but we believe that the GNEPs could be an interesting framework to train adversarial neural networks.

**Outline.** The article has been organized as follows: Section 2 focuses on notations and preliminary work. In particular, working of GNEPs and neuromorphic cameras have been described. Then in section 3, we review the previous contrast maximization methods. In sections 4 and 5 we introduce our model as well as provide results pertaining to existence of solutions. Moreover, in section 6 we provide the experimental set-up for our results. Subsequently, several examples are considered in section 7.

## 2. Preliminaries.

### 2.1. Generalized Nash equilibrium problems (GNEPs).

2.1.1. *Overview.* A Generalized Nash Equilibrium Problem (GNEP) is a mathematical formulation used to find the equilibrium solution in games involving multiple players, each with their own independent decision-making. The problem involves determining a set of strategies for all players such that no player can unilaterally improve their payoff by changing their strategy while taking into account the strategies chosen by the other players. We seek to find a solution that satisfies the objective function of each player while considering the strategies of the other players.

2.1.2. *Notation.* Consider a set of  $j = 1, \dots, N$  agents that have control over  $N_v = \sum_{j=1}^N n_j$  variables where  $n_j$  represents the number of variables that agent  $j$  controls. We denote the variables agent  $j$  controls by the vector  $\mathbf{z}^j \in \mathbb{R}^{n_j}$ . Additionally, we denote the vector of all decision variables as:

$$\mathbf{z} := (\mathbf{z}^1 \quad \mathbf{z}^2 \quad \dots \quad \mathbf{z}^N)^\top \in \mathbb{R}^{N_v}.$$

We extend this notation such that

$$\mathbf{z}^{-j} := (\mathbf{z}^1 \quad \mathbf{z}^2 \quad \dots \quad \mathbf{z}^{j-1} \quad \mathbf{z}^{j+1} \quad \dots \quad \mathbf{z}^N)^\top \in \mathbb{R}^{N_v - n_j}$$

represents the decision variables of all agents except agent  $j$ . The goal of each agent is to minimize their own objective function which we denote by  $J^j(\mathbf{z}^j, \mathbf{z}^{-j})$  while taking into account the strategies of all other players. The feasible strategy set for player  $j$  is denoted by  $\mathbf{Z}^j(\mathbf{z}^{-j})$ . This set captures the strategies available to player  $j$  while considering the constraints imposed by the strategies chosen by the other players. We can then express the minimization problem for each player  $j$  as:

$$\min_{\mathbf{z}^j \in \mathbf{Z}^j(\mathbf{z}^{-j})} J^j(\mathbf{z}^j, \mathbf{z}^{-j}). \quad (1)$$

Our goal is then to solve the above minimization problem to find an optimal solution vector which we now define.

**Definition 2.1** (Generalized Nash Equilibrium [6]). A point  $\bar{\mathbf{z}}$  is said to be a (generalized Nash) equilibrium or a solution to the Generalized Nash Equilibrium Problem (GNEP) if it satisfies the following:

$$\bar{\mathbf{z}} = (\bar{\mathbf{z}}^1 \quad \bar{\mathbf{z}}^2 \quad \dots \quad \bar{\mathbf{z}}^N)^\top$$

such that

$$\bar{z}^j \in Z^j(\bar{z}^{-j}) \quad \text{for all } j = 1, 2, \dots, N.$$

For more information on GNEP's and their solutions we refer the reader to [6, 11, 14, 19].

## 2.2. Neuromorphic cameras.

**2.2.1. Overview.** Neuromorphic cameras consist of individual pixels that independently detect changes in light intensity in real time. These cameras sample light intensity at a rapid rate, typically on the order of microseconds, and record events when the intensity exceeds a preset threshold  $c$ . The use of high-speed sampling and the independent nature of neuromorphic camera pixels make them highly resistant to issues such as overexposure and image blurring. The latter has been recently explored in detail in [1], both from a theoretical and algorithmic point of view.

**2.2.2. Data representation.** The output of a neuromorphic camera is of the form:  $e = (x, y, t, p)$ . If we consider  $k = 1, \dots, N_e$  distinct events, then we can describe each event individually with the following definition:

$$\mathbb{R}^{1 \times 4} \ni \mathbf{e}_k := (\mathbf{x}_k, \mathbf{y}_k, \mathbf{t}_k, \mathbf{p}_k).$$

where subscript  $k$  in  $\mathbf{x}_k$  denotes the  $k$ -th component of vector  $\mathbf{x}$ . Moreover,  $(\mathbf{x}_k, \mathbf{y}_k) \in \Omega = \{1, 2, \dots, N_x\} \times \{1, 2, \dots, N_y\}$  denotes the pixel location of event  $k$  with  $N_x \times N_y$  being the resolution of the camera. Moreover,  $\mathbf{t}_k \in \mathbb{R}$  represents the time at which event  $k$  occurred and  $\mathbf{p}_k \in \{-1, 1\}$  represents the polarity of event  $k$ .

The polarity recorded by a neuromorphic camera at a given pixel  $(\mathbf{x}_k, \mathbf{y}_k)$  is a function of the instantaneous light intensity  $(\mathbf{q}_k)$  recorded at  $(\mathbf{x}_k, \mathbf{y}_k)$  at time  $\mathbf{t}_k$ . Furthermore, the reference light intensity for pixel  $(\mathbf{x}_k, \mathbf{y}_k)$  at time  $\mathbf{t}_k$  is given by  $\mathbf{q}_{ref_k}$ . Using these quantities we define the polarity as:

$$\mathbf{p}_k := \begin{cases} +1, & \log\left(\frac{\mathbf{q}_k}{\mathbf{q}_{ref_k}}\right) \geq c \\ -1, & \log\left(\frac{\mathbf{q}_k}{\mathbf{q}_{ref_k}}\right) \leq -c. \end{cases}$$

No event is recorded if  $\mathbf{p}_k \in (-c, c)$ . Notice that,  $c$  here indicates the camera threshold.

Next, we can describe a collection of  $N_e \in \mathbb{N}$  events with the following definition:

$$\mathbb{R}^{N_e \times 4} \ni \mathbf{e} := (\mathbf{x}, \mathbf{y}, \mathbf{t}, \mathbf{p}).$$

We are interested in whether an event has occurred or not, the positive or negative sign on polarity is not relevant here. Therefore, we will omit the polarity column vector and redefine  $\mathbf{e}$  as:

$$\mathbb{R}^{N_e \times 3} \ni \mathbf{e} := (\mathbf{x}, \mathbf{y}, \mathbf{t}).$$

A visual representation of event data is given in Figure 2.

**2.3. Warping function.** In the context of this paper the warping function refers to a mapping  $\mathbf{W} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ . In general, given some motion parameter  $\boldsymbol{\theta}$  and a reference time  $t_0 \in \mathbb{R}$  we project the events,  $\mathbf{e}$  to this reference time. This process is shown visually in Figures 14, 15 in Appendix A.

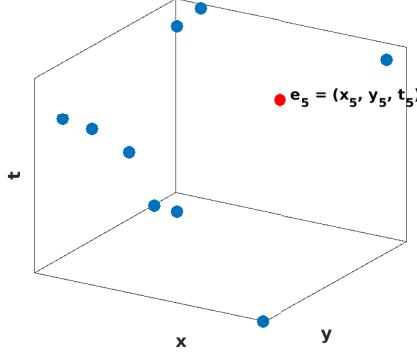


FIGURE 2. In this figure we isolate an event,  $e_5$ , shown as the red dot.

**2.4. Row-wise 2-Norm.** We define the operator  $\|\cdot\|_{2,r} : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^M$  to be a row-wise 2 norm. Given a matrix  $V \in \mathbb{R}^{M \times N}$  we have the following:

$$\|V\|_{2,r} = (\|V_{(1,:)}\|_2 \ \|V_{(2,:)}\|_2 \ \dots \ \|V_{(M,:)}\|_2)^\top.$$

Here  $V_{(M,:)}$  represents the  $M$ -th row of  $V$ .

**3. Prior work.** The approach presented in this paper draws inspiration from the Contrast Maximization (CM) technique, which we now formally introduce. The CM method seeks to find a set of motion parameters  $\{\theta^1, \theta^2, \dots, \theta^N\}$  (one for each object in the scene) that maximizes the contrast in an image of warped events. For a visual interpretation of  $\theta^j$ , we refer to Appendix A. This image of warped events, denoted as  $I \in \mathbb{R}^{N_x \times N_y}$ , is generated by projecting clusters of events onto a reference time,  $t_0 \in \mathbb{R}$ , via a warping function (cf. Appendix A). When the parameter  $\theta^j$  accurately captures the motion of a specific object, the projected events associated with that object coalesce into a distinct and well-defined cluster within the image  $I$ . The level of clarity in this image can be assessed using various functions, such as Image Variance, Magnitude of the Image Gradient, and Image Entropy [8]. The level of clarity serves as a metric for how accurately the motion parameter  $\theta$  approximates the true motion. Furthermore, given a set of motion parameters,  $\{\theta^j\}_{j=1}^N$ , we are then able to perform event based image segmentation by segmenting the events by their respective motion parameters. A general outline for techniques centering on contrast maximization (for a single object) is given by [25]:

1. Estimate/Initialize a motion parameter,  $\theta$ .
2. Use a warping function,  $\mathcal{W}(\theta)$ , to project events to a reference time,  $t_0$ , which generates an image of warped events,  $I$ .
3. Evaluate  $I$  with a reward function.
4. Perform an optimization step with respect to the motion parameter  $\theta$ .

Some care has to be given to initializing  $\theta$  since this problem is inherently non-convex and there are multiple local minima to consider [20]. Previous efforts have utilized prior knowledge such as optical flow information or other initialization techniques as shown in [10, 20, 23, 25]. In scenarios where multiple objects coexist within a scene several challenges arise:

1. The difficulty in initialization of motion parameters is compounded due to multiple objects.
2. When faced with events generated by multiple objects, the task extends beyond identifying motion parameters for each object. It becomes crucial to accurately discern the specific events associated with each object. This results in the need for a clustering step to be added to the contrast maximization outline. [20, 25].
3. Additional prior knowledge may be needed such as number of objects in the scene [20].

Our work is a departure from the above standard approach. We do not use the motion parameter,  $\theta$ , as our optimization variable. Instead we establish a GNEP framework where the motion parameters  $\{\theta^j\}_{j=1}^N$  are formulated as a function of the events and the strategies chosen by each player. We then seek to optimize the strategy of each player. With this framework we are able to avoid the need for prior knowledge in order to initialize the motion parameters and we do not need to perform an alternating optimization method as in [20] and [25]. In these prior methods, two variables had to be optimized, the motion parameters  $\{\theta^j\}_{j=1}^N$  and the cluster associations for each event,  $P_{j,k}$ , with one variable being held fixed while the other is iteratively optimized. Additionally our method does not require any prior knowledge such as optical flow or the number of objects in the scene.

**4. A GNEP based model.** In our model, each player is assigned the task of selecting a strategy that maximizes their own local objective function, denoted as  $J^j$ . The chosen strategy involves assigning a confidence value to each event, where the confidence value assigned to event  $k$  reflects the player's level of certainty that the events with non-zero confidence scores are generated by the same object. An outline for each player is given as follows:

1. Player  $j$  chooses a strategy  $z^j$  while taking into account the strategies of the other players,  $z^{-j}$ . We show an optimal strategy for two players in Figure 13.
2. Calculate the motion parameter for player  $j$ , denoted by  $\theta^j$ , using a least squares approximation. This average motion of events is formally defined in Section 4.2. The slope of the lines shown in Figure 14 represent  $\theta^j$ .
3. Calculate the image of warped events,  $I^j$  based upon the strategy chosen,  $z^j$ . An optimal image of warped events is shown in Figure 15.
4. Evaluate the image of warped events,  $I^j$ , with regard to a reward function.
5. Perform an optimization step with respect to the strategy  $z^j$ .

Next, we describe our model mathematically.

**4.1. Generalized Nash equilibrium model.** Consider a set of  $N$  players such that  $j = 1, \dots, N$  and event data  $e \in \mathbb{R}^{N_e \times 3}$ . We will denote the  $j$ -th player strategy as:

$$z^j \in \mathbf{Z}^j(z^{-j}) := \left\{ z^j \in [0, 1]^{N_e} \mid g(z) \leq \mathbf{0} \right\}. \quad (2)$$

The strategy  $z^j$  represents the confidence values assigned by player  $j$  to indicate their level of confidence in the relationship between the events. Recall  $N_e$  is the total number of events and  $g(z)$  is a set of linear constraints given by:

$$g(z) := \sum_{j=1}^N z^j - \mathbf{b},$$

with

$$\mathbf{b} := (1 \quad 1 \quad \dots \quad 1)^\top \in \mathbb{R}^{N_e}.$$

The linear constraints, represented by the function  $\mathbf{g}(\mathbf{z})$ , impose a restriction on the sum of confidence values assigned to each event. This restriction ensures that the cumulative confidence values for each event do not exceed a maximum value of 1. By incorporating these constraints, we maintain the coherence and integrity of the confidence assignments, enabling a more accurate representation of the associations between events. Next we will begin building our objective function starting with calculating  $\boldsymbol{\theta}$ .

**4.2. Average motion of events.** The average motion of events is the least squares approximation of the change in  $x, y$  directions with respect to time of the chosen strategy. We denote  $\mathbb{H}(s)$  to be the Heaviside step function:

$$\mathbb{H}(s) := \begin{cases} 1, & s > 0 \\ 0, & s \leq 0. \end{cases} \quad (3)$$

We then calculate the change in the  $x$ -direction with respect to time of the events chosen in strategy  $\mathbf{z}^j$  by the following:

$$\theta_x^j := \frac{\mathbb{H}(\mathbf{z}^j)^\top ((\mathbf{t} - \mu_t(\mathbf{z}^j) \cdot \mathbf{b}) \odot (\mathbf{x} - \mu_x(\mathbf{z}^j) \cdot \mathbf{b}))}{\mathbb{H}(\mathbf{z}^j)^\top ((\mathbf{t} - \mu_t(\mathbf{z}^j) \cdot \mathbf{b}) \odot (\mathbf{t} - \mu_t(\mathbf{z}^j) \cdot \mathbf{b}))}. \quad (4)$$

Notice that  $\mathbb{H}$  acts component-wise when applied to vectors. Furthermore  $\mu_t(\mathbf{z}^j)$  and  $\mu_x(\mathbf{z}^j)$  are defined as

$$\mu_t(\mathbf{z}^j) := \frac{\mathbb{H}(\mathbf{z}^j)^\top \mathbf{t}}{\sum_{k=1}^{N_e} \mathbb{H}((\mathbf{z}^j)_k)} \quad \text{and} \quad \mu_x(\mathbf{z}^j) := \frac{\mathbb{H}(\mathbf{z}^j)^\top \mathbf{x}}{\sum_{k=1}^{N_e} \mathbb{H}((\mathbf{z}^j)_k)}. \quad (5)$$

We calculate  $\theta_y^j$  similarly using  $\mathbf{y}$  in place of  $\mathbf{x}$  in Equation (4). This gives us:

$$\boldsymbol{\theta}^j := (\theta_x^j, \theta_y^j).$$

**4.3. Warping function.** We define a velocity based warping function as follows:

$$\mathbb{R}^{N_e \times 2} \ni \mathbf{W}^j(\mathbf{e}, t_0, \mathbf{z}^j) := (\mathbf{x}, \mathbf{y}) + (t_0 \cdot \mathbf{b} - \mathbf{t}) \odot (\theta_x^j \cdot \mathbf{b}, \theta_y^j \cdot \mathbf{b}). \quad (6)$$

Here the symbol  $\odot$  represents element-wise multiplication known as the Hadamard product. Note that we have defined a warping function that describes linear motion. If non-linear motion is expected then this warping function can be updated to accommodate. Next we formally define an image of warped events for each player  $j$ .

**4.4. Image of warped events.** The image of warped events is the result of translating the events chosen by strategy  $\mathbf{z}^j$  to a given time,  $t_0 \in \mathbb{R}$  with the warping function  $\mathbf{W}^j(\mathbf{e}, t_0, \mathbf{z}^j)$ . We describe this process mathematically for a given pixel,  $(\hat{x}, \hat{y}) \in \Omega$ , as follows:

$$\mathbb{R} \ni \mathbf{I}^j((\hat{x}, \hat{y}), t_0, \mathbf{z}^j) := (\mathbf{z}^j)^\top \delta \left( \|\mathbf{W}^j(\mathbf{e}, t_0, \mathbf{z}^j) - (\hat{x} \cdot \mathbf{b}, \hat{y} \cdot \mathbf{b})\|_{2,r} \right). \quad (7)$$

Here  $\|\cdot\|_{2,r}$  is the row-wise 2 norm defined in Section 2.4 and  $\delta$  refers to the Kronecker delta centered at 0 defined as:

$$\delta(\gamma) = \begin{cases} 0, & \gamma \neq 0 \\ 1, & \gamma = 0; \end{cases} \quad (8)$$



and is applied elementwise to vectors. We describe the full image of warped events as:

$$\mathbf{I}^j(\boldsymbol{\Omega}, t_0, \mathbf{z}^j) \in \mathbb{R}^{N_x \times N_y}.$$

For the remainder of the paper we will omit  $t_0$  from  $\mathbf{I}^j$  and  $\mathbf{W}^j$  for ease of notation.

**4.5. Objective function.** Finally, we define our objective function  $J^j$  as:

$$J^j(\mathbf{z}^j) := -E(\mathbf{I}^j(\boldsymbol{\Omega}, \mathbf{z}^j)) + \frac{\lambda_1}{2} \left( V(\mathbf{z}^j, \mathbf{x}) + V(\mathbf{z}^j, \mathbf{y}) \right) + \frac{\lambda_2}{2} \|\mathbb{H}(\mathbf{z}^j)\|_2^2. \quad (9)$$

Where  $E(\mathbf{I}^j)$  is a modified entropy function given by:

$$E(\mathbf{I}^j) = \sum_{\hat{x}=1}^{N_x} \sum_{\hat{y}=1}^{N_y} \mathbf{I}^j((\hat{x}, \hat{y}), \mathbf{z}^j) \cdot \log \left( \mathbf{I}^j((\hat{x}, \hat{y}), \mathbf{z}^j) \right), \quad (10)$$

and

$$V(\mathbf{z}^j, \mathbf{x}) := \frac{1}{N_x} \left( \left( \mathbb{H}(\mathbf{z}^j) \odot \mathbf{x} \right) - \mu_x(\mathbf{z}^j) \cdot \mathbf{b} \right)^\top \left( \left( \mathbb{H}(\mathbf{z}^j) \odot \mathbf{x} \right) - \mu_x(\mathbf{z}^j) \cdot \mathbf{b} \right). \quad (11)$$

Similarly, we define  $V(\mathbf{z}^j, \mathbf{y})$ . Notice that  $V(\mathbf{z}^j, \cdot)$  denotes the variance penalty term with  $\lambda_1 \geq 0$  being the penalty parameter. A larger  $\lambda_1$  penalizes associating events that are spatially further apart. Tuning of this parameter allows us to differentiate between objects that have similar motion parameters but are spatially distinct in the image. Furthermore we have the regularization term  $\frac{\lambda_2}{2} \|\mathbb{H}(\mathbf{z}^j)\|_2^2$  with  $\lambda_2 \geq 0$ . This term serves to limit the number of extraneous events (noise) chosen by each player. Then we have the following model for each player,  $j = 1, \dots, N$ :

$$\begin{aligned} \min \quad & J^j(\mathbf{z}^j), \\ \text{subject to} \quad & \mathbf{z}^j \in \mathbf{Z}^j(\mathbf{z}^{-j}), \end{aligned} \quad (12)$$

where  $\mathbf{Z}^j(\mathbf{z}^{-j})$  is defined in (2).

Since the objective functions,  $J^j(\mathbf{z}^j)$ , are not convex we are not able to solve (12) directly using the standard approach of quasi-variational inequalities. Instead we rely on leader / follower methods in the form of a  $N$ -level optimization problem to search for equilibrium of (12), where  $N$  is the number of players [5, C.3, S.4, pp.68-73]. We present the  $N$ -level optimization problem as:

$$\min_{\mathbf{z}^N \in \mathbf{Z}^N(\mathbf{z}^{-N})} J^N(\mathbf{z}^N) \quad (13)$$

subject to

$$\mathbf{z}^j = \operatorname{argmin} \left\{ J^j(\mathbf{z}^j) \mid \sum_{i=1}^j \mathbf{z}^i \leq \mathbf{b}, \mathbf{z}^j \in [0, 1]^{N_e} \right\} \quad \text{for } j = 1, \dots, N-1.$$

We notice a subtle difference between (13) and (12). Each player in (12) has complete information of all other player strategies. While in (13) each player only takes the strategy of the previous players into account. For example, for player 2 we have the following problem:

$$\mathbf{z}^2 = \operatorname{argmin} \left\{ J^2(\mathbf{z}^2) \mid \sum_{i=1}^2 \mathbf{z}^i \leq \mathbf{b}, \mathbf{z}^2 \in [0, 1]^{N_e} \right\}.$$

Expanding the constraint we get:

$$\sum_{i=1}^2 \mathbf{z}^i = \mathbf{z}^1 + \mathbf{z}^2.$$

Hence we must first calculate  $\mathbf{z}^1$  before calculating  $J^2$ . Notice that, the player  $N$  accounts for the strategies of players  $1, \dots, N-1$ . For this reason, problems from (13) will need to be solved in ascending order starting with player 1. We note that we solve this problem in a single pass.

---

**Algorithm 1** N-level Optimization Algorithm

---

**Input:**  $e$  - Event data,  $\lambda_1, \lambda_2$  - Penalty parameters

---

- 1: **for**  $j = 1$  to  $N$  **do**
  - 2:     Initialize  $\mathbf{z}_{init}^j$
  - 3:     Calculate  $\bar{\mathbf{z}}^j$  shown in Equation (13)      $\triangleright \mathbf{z}_{init}^j$  is used as the initial value.
  - 4: **end for**
  - 5: **Return**  $\bar{\mathbf{z}} = \begin{pmatrix} \bar{\mathbf{z}}^1 & \bar{\mathbf{z}}^2 & \dots & \bar{\mathbf{z}}^N \end{pmatrix}^\top$   $\triangleright$  This is the approximation to solution in Def. 2.1.
- 

In the next section we show the existence of solutions for (13), and that the set of solutions for  $j = 1, \dots, N$  corresponds to an equilibrium of (12).

## 5. Analysis of the proposed model.

**5.1. Existence of equilibrium.** To establish the validity of the  $N$ -level optimization presented in Equation (13) as a method for computing an equilibrium to (12), we begin by demonstrating the existence of a solution. This process involves several steps, with the initial focus on establishing the finiteness of the set  $\{\mathbb{H}(\mathbf{z}^j) \mid \mathbf{z}^j \in [0, 1]^{N_e}\}$ .

**Lemma 5.1.** *The set  $\{\mathbb{H}(\mathbf{z}^j) \mid \mathbf{z}^j \in [0, 1]^{N_e}\}$  is finite.*

*Proof.* For ease of notation we omit the superscript  $j$  from  $\mathbf{z}$ . We notice that  $\mathbb{H}(\mathbf{z}) \in \{0, 1\}^{N_e}$  for all  $\mathbf{z}$ , thus  $\{\mathbb{H}(\mathbf{z}) \mid \mathbf{z} \in [0, 1]^{N_e}\} \subseteq \{0, 1\}^{N_e}$ . Furthermore the set  $\{0, 1\}^{N_e}$  has cardinality of  $2^{N_e}$  and is therefore finite. Hence  $\{\mathbb{H}(\mathbf{z}) \mid \mathbf{z} \in [0, 1]^{N_e}\}$  is finite.  $\square$

Next we show that the quantities  $V(\mathbf{z}, \cdot)$  and  $V(\mathbb{H}(\mathbf{z}), \cdot)$  are equal.

**Lemma 5.2.** *For any  $\mathbf{z} \in \mathbb{R}^{N_e}$  we have  $V(\mathbf{z}, \cdot) = V(\mathbb{H}(\mathbf{z}), \cdot)$ .*

*Proof.* This follows directly from the definition of  $V$  in (11).  $\square$

We are now ready to establish existence of solution to (13).

**Theorem 5.3.** *There exists a solution to each level of the optimization problem (13).*

*Proof.* For ease of notation we omit the superscript  $j$ .

Part I: Single player case. We focus on player 1 as the result will directly apply to other players. First we observe for any  $\mathbf{z} \in [0, 1]^{N_e}$

$$\mathbb{H}(\mathbf{z}) \leq \mathbf{b}$$

by definition of the Heaviside function given in Equation (3). Next we note from Lemma 5.1 that the set  $\{\mathbb{H}(\mathbf{z}) \mid \mathbf{z} \in [0, 1]^{N_e}\}$  is finite. Since the set is finite we know there exists some  $\hat{\mathbf{z}} \in [0, 1]^{N_e}$  such that  $\bar{\mathbf{z}} := \mathbb{H}(\hat{\mathbf{z}})$  fulfills

$$J(\bar{\mathbf{z}}) \leq J(\mathbb{H}(\mathbf{z})) \quad \text{for all } \mathbf{z} \in [0, 1]^{N_e}. \quad (14)$$

Here  $\bar{\mathbf{z}}$  is our candidate for a solution to (13). Next we show for arbitrary  $\mathbf{z}$  that  $J(\mathbb{H}(\mathbf{z})) \leq J(\mathbf{z})$ . If  $\mathbf{z} = \mathbb{H}(\mathbf{z})$  then we are done. Else, there exist some  $k = 1, \dots, N_e$  such that  $0 < z_k < 1$ . Then for any  $(\hat{x}, \hat{y}) \in \Omega$  and setting

$$\boxed{\mathbf{I}} := \|\mathbf{W}(\mathbf{e}, \mathbf{z}) - (\hat{x} \cdot \mathbf{b}, \hat{y} \cdot \mathbf{b})\|_{2,r}$$

we immediately obtain that

$$z_k \cdot \delta \left( \boxed{\mathbf{I}} \right)_k \leq \mathbb{H}(z_k) \cdot \delta \left( \boxed{\mathbf{I}} \right)_k, \quad (15)$$

where  $\delta \left( \boxed{\mathbf{I}} \right)_k$  represents the  $k$ -th element of  $\delta \left( \boxed{\mathbf{I}} \right)$ . Using Equation (15) and the fact that  $\log$  is a monotone increasing function we have the following inequality:

$$z_k \cdot \delta \left( \boxed{\mathbf{I}} \right)_k \cdot \log \left( z_k \cdot \delta \left( \boxed{\mathbf{I}} \right)_k \right) \leq \mathbb{H}(z_k) \cdot \delta \left( \boxed{\mathbf{I}} \right)_k \cdot \log \left( \mathbb{H}(z_k) \cdot \delta \left( \boxed{\mathbf{I}} \right)_k \right). \quad (16)$$

We note the common definition of  $0 \cdot \log(0) := 0$ , hence this holds even for  $\delta \left( \boxed{\mathbf{I}} \right)_k = 0$ . Using the definition of  $\mathbf{I}(\Omega, \cdot)$  in (7) and Equation (16) we have:

$$\mathbf{I}((\hat{x}, \hat{y}), \mathbf{z}) \leq \mathbf{I}((\hat{x}, \hat{y}), \mathbb{H}(\mathbf{z})).$$

Taking the sum over all possible  $(\hat{x}, \hat{y}) \in \Omega$  gives us:

$$E(\mathbf{I}(\Omega, \mathbf{z})) \leq E(\mathbf{I}(\Omega, \mathbb{H}(\mathbf{z}))). \quad (17)$$

Combining Equations (14) and (17) along with Lemma 5.2 we have for any  $\mathbf{z} \in [0, 1]^{N_e}$ :

$$J(\mathbf{z}) \geq J(\mathbb{H}(\mathbf{z})) \geq J(\bar{\mathbf{z}}).$$

We conclude that a solution to (13) exists for a single player.

Part II: Two player case. We now show that given

$$\bar{\mathbf{z}}^1 = \operatorname{argmin} \left\{ J^1(\mathbf{z}^1) \mid \mathbf{z}^1 \leq \mathbf{b}, \mathbf{z}^1 \in [0, 1]^{N_e} \right\},$$

a solution exists for

$$\min \left\{ J^2(\mathbf{z}^2) \mid \sum_{i=1}^2 \mathbf{z}^i \leq \mathbf{b}, \mathbf{z}^2 \in [0, 1]^{N_e} \right\}.$$

First we notice from Part I that the components of  $\bar{\mathbf{z}}^1$  are either 0 or 1. Now, define the set

$$M = \{m = 1, \dots, N_e \mid (\bar{\mathbf{z}}^1)_m = 1\} \subset \mathbb{R}^{\#M}.$$

Then for  $m \in M^c$  we have  $(\bar{\mathbf{z}}^1)_m = \mathbf{0} \in \mathbb{R}^{\#M^c}$ . Since we have

$$\mathbf{b} \geq \sum_{i=1}^2 \mathbf{z}^i = \begin{pmatrix} (\bar{\mathbf{z}}^1)_{m \in M} + (\bar{\mathbf{z}}^2)_{m \in M} \\ (\bar{\mathbf{z}}^1)_{m \in M^c} + (\bar{\mathbf{z}}^2)_{m \in M^c} \end{pmatrix}.$$

Using the definition of  $\mathbf{b}$ , for  $m \in M$ , we have that  $(\bar{\mathbf{z}}^2)_m = \mathbf{0} \in \mathbb{R}^{\#M}$  and for  $m \in M^c$  we have that  $(\bar{\mathbf{z}}^2)_m \in [0, 1]^{\#M^c}$ . This proves the existence of  $\bar{\mathbf{z}}^2$ . Similar arguments helps prove existence of  $\bar{\mathbf{z}}^3$ , etc. and the proof is complete.  $\square$

Next, we show that the  $N$ -level optimization presented in Equation (13) produces an equilibrium point to (12).

**Theorem 5.4.** *The  $N$ -level optimization described in (13) produces a Generalized Nash Equilibrium to (12).*

*Proof.* Let  $\bar{z}^j$ ,  $j = 1, \dots, N$ , be the resulting argmin from each level of (13). Then we have the following:

$$\sum_{j=1}^N \bar{z}^j \leq \mathbf{b} \implies \sum_{j=1}^N \bar{z}^j - \mathbf{b} \leq \mathbf{0}.$$

Furthermore, from (13) we know  $\bar{z}^j \in [0, 1]^{N_e}$ . Then for each  $j$  we have

$$\bar{z}^j \in \left\{ [0, 1]^{N_e} \mid \sum_{i=1}^N \bar{z}^i - \mathbf{b} \leq \mathbf{0} \right\} =: \mathbf{Z}^j(\bar{z}^{-j}).$$

We conclude by Definition 2.1  $\bar{z} = (\bar{z}^1 \quad \bar{z}^2 \quad \dots \quad \bar{z}^N)^\top$  is an equilibrium point of (12).  $\square$

## 6. Experimental setup.

**6.1. Event data pre-processing.** We observe that the dimension of  $\mathbf{z}^j$  directly corresponds to the number of events under consideration. As the number of events increases, so does the dimensionality of the variable, potentially leading to computational challenges. To mitigate this issue, we employ a pre-processing step where we selectively remove time intervals of events from our data. This is accomplished by breaking the time interval into a series of sub intervals, and then removing every other interval from the dataset. By doing so, we effectively limit the dimensionality of the optimization variable, allowing for more efficient and manageable computations while still preserving the essential information contained within the dataset. Figure 3 gives a visual example.

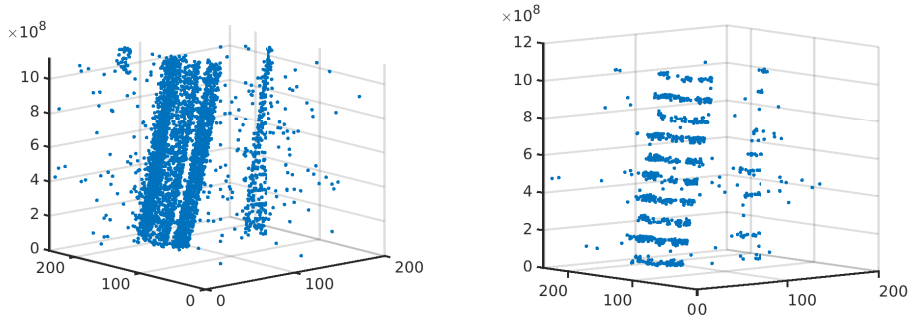


FIGURE 3. In this figure we see the original event data in the left panel and the pre-processed data in the right panel.

**6.2. Numerical relaxations.** We make several numerical relaxations in order to allow for gradient based methods.

6.2.1. *Heaviside relaxation.* First we relax the Heaviside function described in (3) to a continuous approximation given by:

$$\hat{\mathbb{H}}(\mathbf{z}^j) := \frac{1}{2}(1 + \tanh(m \cdot (\mathbf{z}^j - n \cdot \mathbf{b}))).$$

Figure 4 gives a visual representation of the approximation. For the purposes of this paper we chose  $m = 25$  and  $n = 0.25$  for each of our examples. Through testing we found for  $m > 50$  that the gradient of  $\hat{\mathbb{H}}(\mathbf{z}^j)$  becomes unstable. We see this visually in Figure 5. Furthermore, we choose  $n > 0$  to ensure  $\hat{\mathbb{H}}(0) \approx 0$  and  $\hat{\mathbb{H}}(1) \approx 1$ . If this is not the case, then the quality of the results decreases significantly.

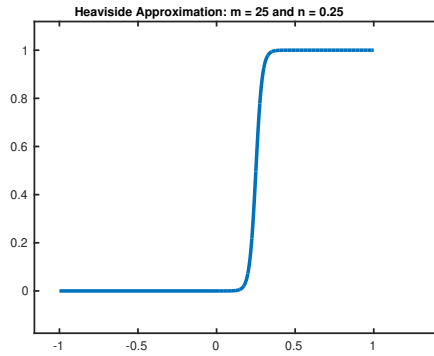


FIGURE 4. In this figure give a visual representation of the Heaviside relaxation function  $\hat{\mathbb{H}}(\mathbf{z})$  (for  $\mathbf{z} \in [-1, 1]$ )

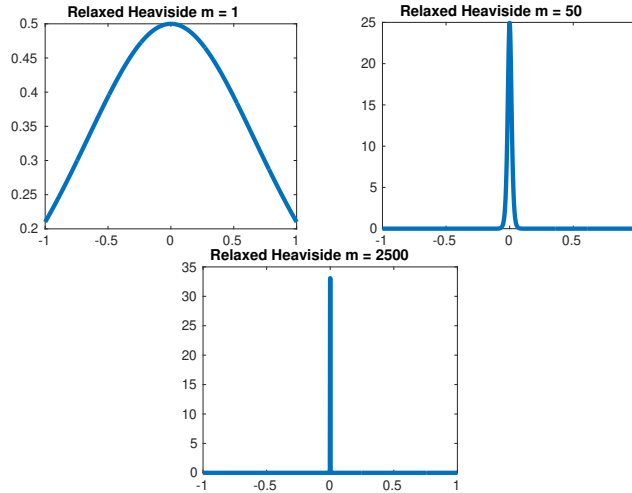


FIGURE 5. In this figure we plot  $\nabla \hat{\mathbb{H}}(\mathbf{z}^j)$  for  $m = 1, 50$ , and  $2500$ . As  $m$  increases,  $\nabla \hat{\mathbb{H}}(\mathbf{z}^j)$  approximates an impulse which can be unstable in numerical calculations.

6.2.2. *Kronecker Delta relaxation.* We also use the function:

$$\exp(-\gamma \|\cdot\|_2^2)$$

to approximate the Kronecker delta  $\delta(\|\cdot\|_2)$  defined in (8). Here  $\gamma$  is chosen to be large enough such that the bump function decays rapidly. For the purposes of this paper we choose  $\gamma = 30$ . For  $\gamma < 5$  there can be a drop off in the quality of results due to the bump function significantly overlapping more than one pixel. This can be seen numerically in the following example:  $\exp(-2 \cdot \|(1,1) - (1,2)\|_2^2) = 0.1353$ . We notice that (1,1) and (1,2) are one pixel apart, yet because  $\gamma$  is small  $0 \ll \exp(-2 \cdot \|(1,1) - (1,2)\|_2^2)$ , i.e., we have a poor approximation of Kronecker delta. Figure 6 gives a visual representation of the relaxation function with  $\gamma = 30$ .

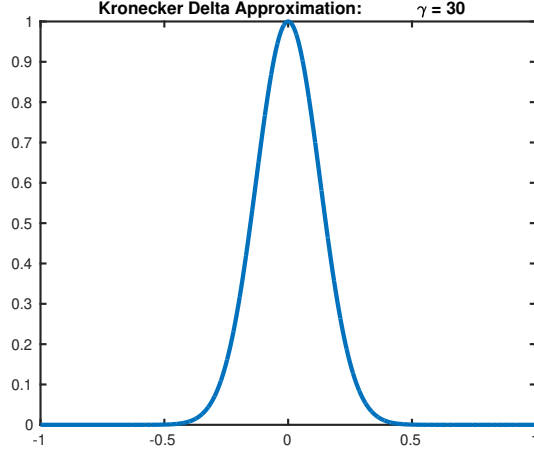


FIGURE 6. In this figure give a visual representation of the Kronecker Delta relaxation function  $\exp(-\gamma \cdot \|z\|_2^2)$  (for  $z \in [-1, 1]$ )

6.2.3. *Entropy relaxation.* We relax the Entropy function  $E$  defined in (10) by introducing two scalars parameters  $\alpha$  and  $\beta$ :

$$E(I^j) = \sum_{\hat{x}=1}^{N_x} \sum_{\hat{y}=1}^{N_y} I^j((\hat{x}, \hat{y}), z^j) \cdot \log \left( (\alpha) \cdot I^j((\hat{x}, \hat{y}), z^j) + \beta \right).$$

These parameters are introduced in order to avoid  $\log(0)$  numerically. We must impose the constraint:  $\alpha = 1 - \beta$  to ensure  $\log(\alpha + \beta) = 0$ . Under this condition the theoretical proofs in section 5 hold. We only check one of the key inequalities (16) next.

**Lemma 6.1.** *If  $\alpha + \beta = 1$  where  $\alpha, \beta \in (0, 1]$  then the inequality (16) becomes*

$$z \cdot \delta(\|\cdot\|^2) \cdot \log \left( (\alpha) \cdot z \cdot \delta(\|\cdot\|^2) + \beta \right) \leq \mathbb{H}(z) \cdot \delta(\|\cdot\|^2) \cdot \log \left( (\alpha) \cdot \mathbb{H}(z) \cdot \delta(\|\cdot\|^2) + \beta \right)$$

for any  $z \in [0, 1]$ .

*Proof.* Let  $\alpha + \beta = 1$ . Then for  $z = 0$  or  $z = 1$  we have  $z = \mathbb{H}(z)$  and we are done. Furthermore if  $\delta(\|\cdot\|^2) = 0$  we get equality and we are done. Now, suppose

$0 < z < 1$  and  $\delta(\|\cdot\|^2) = 1$ . Then we have the following:

$$z \cdot \delta(\|\cdot\|^2) \cdot \log \left( (\alpha) \cdot z \cdot \delta(\|\cdot\|^2) + \beta \right) = z \cdot \log \left( (\alpha) \cdot z + \beta \right) < \log(\alpha + \beta) = 0.$$

Furthermore we have:

$$\mathbb{H}(z) \cdot \delta(\|\cdot\|^2) \cdot \log \left( (\alpha) \cdot \mathbb{H}(z) \cdot \delta(\|\cdot\|^2) + \beta \right) = 0.$$

We conclude

$$z \cdot \delta(\|\cdot\|^2) \cdot \log \left( (\alpha) \cdot z \cdot \delta(\|\cdot\|^2) + \beta \right) \leq \mathbb{H}(z) \cdot \delta(\|\cdot\|^2) \cdot \log \left( (\alpha) \cdot \mathbb{H}(z) \cdot \delta(\|\cdot\|^2) + \beta \right),$$

given  $\alpha + \beta = 1$ . □

### 6.3. Solvers: Genetic algorithm and SQP algorithm.

**6.3.1. Initialization using genetic algorithm.** For our examples, we first use the MATLAB genetic algorithm, `ga`, to calculate an initial value for use in a gradient based method. Recall in Theorem 5.3 we showed that  $\min J^j(\mathbf{z}^j)$  will occur at a boundary point,  $\mathbb{H}(\mathbf{z})$ . For this reason, we give the Genetic Algorithm integer constraints of 0 and 1. Since the Genetic Algorithm is not a gradient method we do not use the function relaxations given in Section 6.2.

**6.3.2. SQP algorithm.** Once the Genetic Algorithm has completed we then use the output as input to the MATLAB ‘`sqp`’ algorithm which is part of `fmincon`. Since this is a gradient based method the relaxations shown in Section 6.2 are used.

**7. Numerical examples.** Several examples of our method are shown. For each of these results, a combination of a Genetic Algorithm and SQP Algorithm as described in section 6.3 were used. The tolerances for these methods were  $10^{-2}$  and  $10^{-4}$ , respectively. All data used in these examples is freely available via the PROPHESEE website.

In Figure 7, 2016 events are used. The original (unprocessed) event data is shown in the left pane while the image of warped events for both players is shown in the right pane. As shown our method is able to fully segment the image into two distinct objects with player 1 associating events generated by the vehicle and player 2 associating events to the pedestrian. These results for player 1 required 447 iterations of a genetic algorithm and an additional 14 iterations from the SQP algorithm. In addition, the results for player 2 required 366 genetic algorithm iterations and 27 SQP iterations. Furthermore, for player 1, Figure 8, compares the output of the genetic algorithm to the output of the SQP algorithm. As shown in Figure 8 the genetic algorithm does associate like events. The drawback being that there are events that should be associated that are left off. In contrast the SQP algorithm associates many more like events. Moreover, for this particular example the tolerance for each algorithm has been set to  $10^{-4}$  for a fair comparison.

Figures 9 - 10 show our method applied to a multi-vehicle scenario. In Figure 9, 1490 events were used in the model. As shown (right) our model is able to segment the image into two distinct vehicles. For player 1 these results required 211 iterations of the genetic algorithm and an additional 8 iterations from the SQP algorithm. For the second player, 97 iterations of the genetic algorithm and 4 iterations of the SQP algorithm were required.

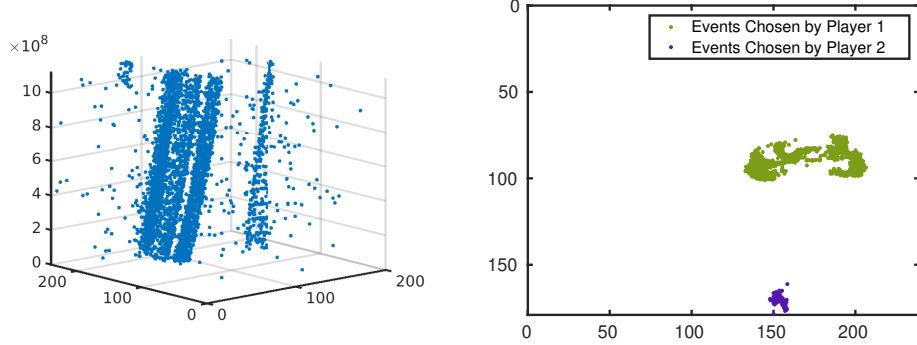


FIGURE 7. In this figure we show the original data in the leftmost pane. The results of our method with two players is shown in the right pane. In this example  $N_e$ , the number of total events, is 2016,  $\lambda_1 = 10^{-3}$ ,  $\lambda_2 = 1$ ,  $\beta = 0.1$  and  $\alpha = 0.9$ .

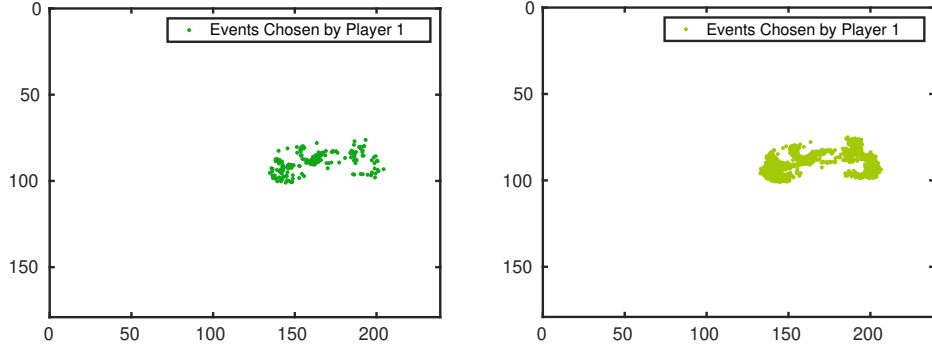


FIGURE 8. In this figure we show the the result of the genetic algorithm in the leftmost pane. In the right pane we show the result of the SQP algorithm, using the strategy produced by the genetic algorithm as an initial value. Notice that no relaxation of Heaviside function or Kronecker delta is used in case of the genetic algorithm. This example motivates that the results with relaxation are meaningful.

Figure 10 shows an additional multi-vehicle scenario with  $N = 2$  players. For this example 2132 events were used. As shown in the right pane, our model is able to segment the image with only two pixels being classified incorrectly. These results required 211 iterations of the genetic algorithm and an additional 14 iterations of the SQP algorithm for player 1. Player 2 required 76 iterations of the genetic algorithm and 8 iterations of the SQP method.

As shown in Figure 11 we have  $N = 4$  players. These results used 1964 events with algorithm iterations shown in Table 1. Left pane shows the processed input data, while the right pane shows our result. We note that there is slight overlap between a few pixels of players 1 and 4 and players 2 and 4 in this result.



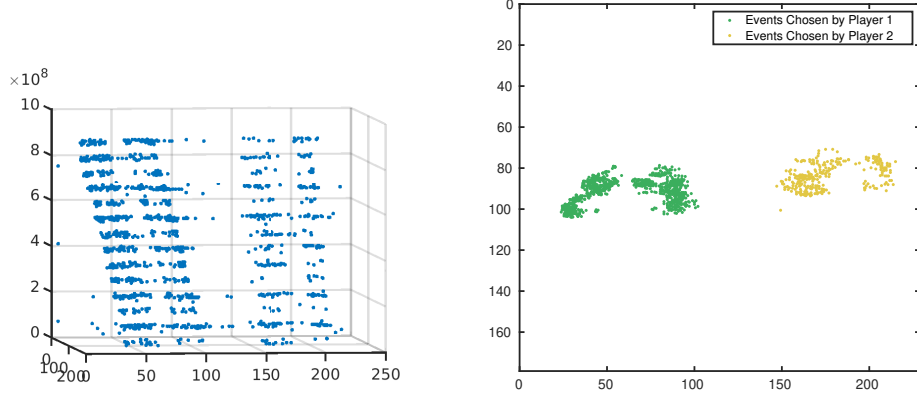


FIGURE 9. In this figure we show the processed data in the leftmost pane. The results of our method with two players is shown in the right pane. In this example  $N_e$ , the number of total events, is 1490,  $\lambda_1 = 10^{-3}$ ,  $\lambda_2 = 10$ ,  $\beta = 0.1$  and  $\alpha = 0.9$ .

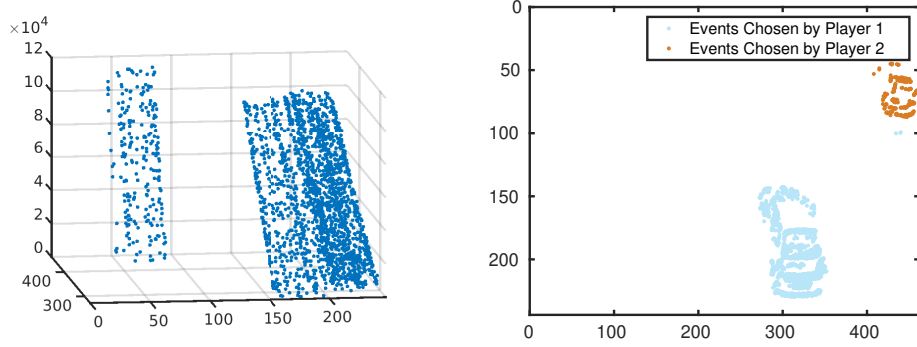


FIGURE 10. In this figure we show the processed data in the leftmost pane. The results of our method with two players is shown in the right pane. In this example  $N_e$ , the number of total events, is 2132,  $\lambda_1 = 10^{-3}$ ,  $\lambda_2 = 1.1$ ,  $\beta = 0.1$  and  $\alpha = 0.9$ .

TABLE 1. Algorithm Iterations for each player in the 4 player example.

	Genetic Algorithm	SQP
Player 1	583	7
Player 2	455	14
Player 3	402	2
Player 4	269	2

**Conclusion, limitation, and future work.** In conclusion, this paper proposes a game theoretic model for the approximation of object velocities and segmentation of event data from a neuromorphic camera. The experimental results demonstrate the effectiveness of the proposed model in achieving accurate and robust segmentation of the event data. Furthermore, we have shown existence of solution to the Generalized

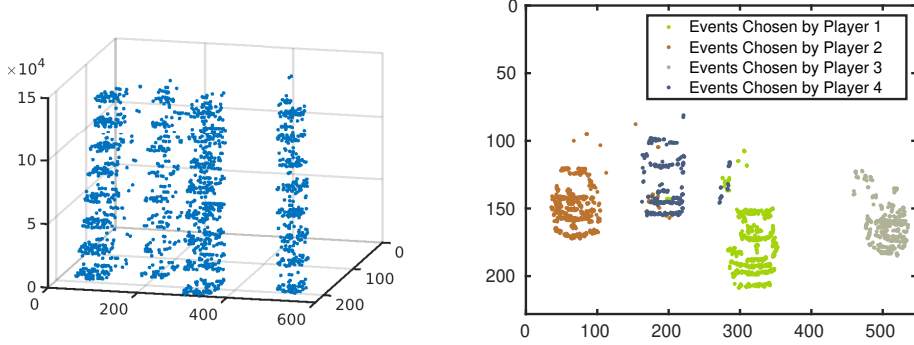


FIGURE 11. In this figure we show the processed data in the left-most pane. The results of our method with four players is shown in the right pane. In this example  $N_e$ , the number of total events, is 1964,  $\lambda_1 = 10^{-3}$ ,  $\lambda_2 = 0$ ,  $\beta = 0.1$  and  $\alpha = 0.9$ .

Nash Equilibrium Problem and we have proposed a  $N$ -level optimization method for calculating equilibrium.

Though our method is able to generate equilibrium there are some limitations. For instance, manual tuning of  $\lambda_1$ ,  $\lambda_2$ , and the highly nonconvex nature of the objective function leads to challenges for initial guesses for the iterates of gradient based methods. In addition, we need to solve  $N$ -optimization problems. Finally, the dimension of the optimization variable  $\mathbf{z}$  is equivalent to the number of events considered this leads to protracted run times for both the Genetic Algorithm and SQP algorithm's. Both of which can take in the between tens of minutes and single digit hours depending on the number of events considered.

In future work we look to explore techniques that will allow us to use the convex hull of the objective function in order to use quasi-variational techniques to calculate equilibrium. Furthermore, we look to incorporate more advanced equations of motion for our warping function.

**Acknowledgments.** The authors are grateful to Prof. Uday Shanbhag (Penn State) for inspiring us to explore the GNEP research area through various discussions and also sharing various references on this topic. We are also thankful to Dr. Patrick O'Neil and Dr. Diego Torrejon (BlackSky) and Dr. Noor Qadri (US Naval Research Lab, Washington DC), for bringing the neuromorphic imaging topic to their attention and for several fruitful discussions.

**Appendix A. Visual representation of our method.** Figure 12 illustrates the event data for two circular events over five timesteps. Player 1 and 2 respectively selects the right and left events. This is shown in Figure 13. A projection mechanism, from 3D to 2D along the lines with slopes  $\theta^1$  and  $\theta^2$ , is shown in Figure 14. The projected data itself is given in Figure 15.

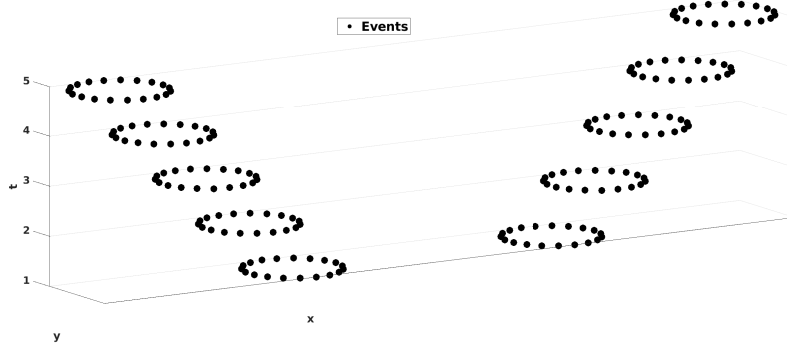


FIGURE 12. In this figure we plot simulated event data of two circular objects over five timesteps.

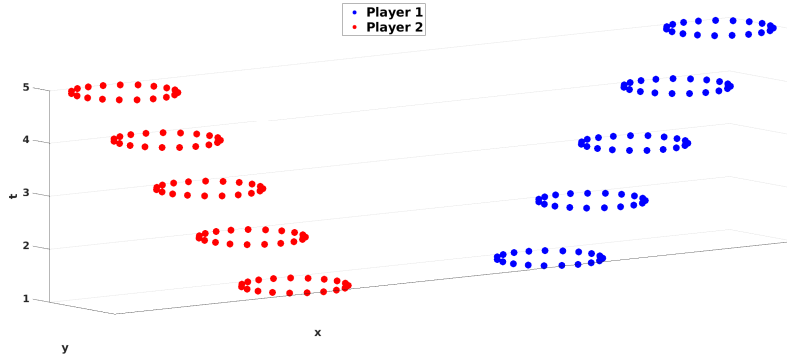


FIGURE 13. In this figure we give a visual representation of the ideal strategies of Player 1 (blue) and Player 2 (red), respectively for the event data given in Figure 12.

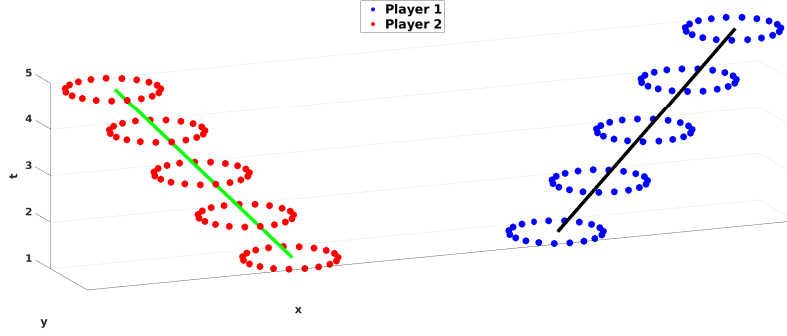


FIGURE 14. In this figure we show the events chosen by Player 1 and Player 2 given in blue and red, respectively. Additionally, we plot projection lines given in black for player 1 and in green for player 2 that have slopes  $\theta^1$  and  $\theta^2$  respectively.

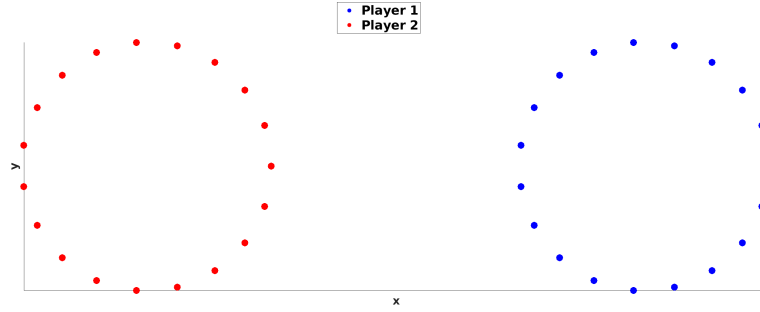


FIGURE 15. In this figure we show the result of our method when applied to the data shown in Figure 12 with strategies of Player 1 and Player 2 shown in Figure 13.

## REFERENCES

- [1] H. Antil and D. Sayre, Bilevel inverse problems in neuromorphic imaging, *Inverse Problems*, **39** (2023), 094003, 22 pp.
- [2] D. Aussel and J. Dutta, [Generalized nash equilibrium problem, variational inequality and quasiconvexity](#), *Operations Research Letters*, **36** (2008), 461-464.
- [3] D. Aussel and A. Svensson, [A short state of the art on multi-leader-follower games](#), *Bilevel Optimization: Advances and Next Challenges*, (2020), 53-76.
- [4] S. Cui and U. V. Shanbhag, [On the computation of equilibria in monotone and potential stochastic hierarchical games](#), *Math. Program.*, **198** (2023), 1227-1285.

- [5] S. Dempe and A. Zemkoho, *Bilevel Optimization*, Springer Optimization and Its Applications, 161, 2020.
- [6] F. Facchinei and C. Kanzow, *Generalized nash equilibrium problems*, *Annals of Operations Research*, **175** (2010), 177-211.
- [7] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, et al., *Event-based vision: A survey*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44** (2020), 154-180.
- [8] G. Gallego, M. Gehrig and D. Scaramuzza, *Focus is all you need: Loss functions for event-based vision*, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 12280-12289.
- [9] G. Gallego, H. Rebecq and D. Scaramuzza, *A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation*, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 3867-3876.
- [10] A. Glover and C. Bartolozzi, *Event-driven ball detection and gaze fixation in clutter*, In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, 2203-2208.
- [11] P. T. Harker, *Generalized nash games and quasi-variational inequalities*, *European Journal of Operational Research*, **54** (1991), 81-94.
- [12] M. Hu and M. Fukushima, *Multi-leader-follower games: Models, methods and applications*, *Journal of the Operations Research Society of Japan*, **58** (2015), 1-23.
- [13] L. A. Julien, *Stackelberg games*, In *Handbook of Game Theory and Industrial Organization, Volume I*, Edward Elgar Publishing, 2018, 261-311.
- [14] A. A. Kulkarni and U. V. Shanbhag, *A shared-constraint approach to multi-leader multi-follower games*, *Set-Valued and Variational Analysis*, **22** (2014), 691-720.
- [15] A. A. Kulkarni and U. V. Shanbhag, *An existence result for hierarchical stackelberg v/s stackelberg games*, *IEEE Transactions on Automatic Control*, **60** (2015), 3379-3384.
- [16] D. Liu, A. Parra and T.-J. Chin, *Globally optimal contrast maximisation for event-based motion estimation*, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 6349-6358.
- [17] J.-S. Pang and M. Fukushima, *Quasi-variational inequalities, generalized nash equilibria, and multi-leader-follower games*, *Computational Management Science*, **6** (2009), 373-375.
- [18] C. M. Parameshwara, S. Li, C. Fermüller, N. J. Sanket, M. S. Evanusa and Y. Aloimonos, *Spikems: Deep spiking neural network for motion segmentation*, In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, 3414-3420.
- [19] D. A. Schiro, J.-S. Pang and U. V. Shanbhag, *On the solution of affine generalized nash equilibrium problems with shared constraints by lemke's method*, *Mathematical Programming*, **142** (2013), 1-46.
- [20] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman and D. Scaramuzza, *Event-based motion segmentation by motion compensation*, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 7244-7253.
- [21] T. Stoffregen and L. Kleeman, *Simultaneous optical flow and segmentation (sofas) using dynamic vision sensor*, arXiv preprint, [arXiv:1805.12326](https://arxiv.org/abs/1805.12326), 2018.
- [22] T. Stoffregen and L. Kleeman, *Event cameras, contrast maximization and reward functions: An analysis*, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 12300-12308.
- [23] V. Vasco, A. Glover, E. Mueggler, D. Scaramuzza, L. Natale and C. Bartolozzi, *Independent motion detection with event-driven cameras*, In *2017 18th International Conference on Advanced Robotics (ICAR)*, IEEE, 2017, 530-536.
- [24] J. Wang, W. Guan, Y. Huang, R. Schober and X. You, *Distributed optimization of hierarchical small cell networks: A gneep framework*, *IEEE Journal on Selected Areas in Communications*, **35** (2017), 249-264.
- [25] Y. Zhou, G. Gallego, X. Lu, S. Liu and S. Shen, *Event-based motion segmentation with spatio-temporal graph cuts*, *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [26] A. Z. Zhu, N. Atanasov and K. Daniilidis, *Event-based feature tracking with probabilistic data association*, In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, 4465-4470.

- [27] A. Z. Zhu, L. Yuan, K. Chaney and K. Daniilidis, [Unsupervised event-based optical flow using motion compensation](#), In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

Received October 2023; 1st revision October 2023; 2nd revision May 2024; early access May 2024.