Extracting synthetic sensors: A study in gap navigation

Chengyuan Qian and Dylan A. Shell

Dept. of Computer Science and Engineering, Texas A&M University.
{cyqian,dshell}@tamu.edu

Abstract. We consider the problem of sensing and navigation using depth discontinuities, commonly called *gaps*. This paper establishes a new theoretical framework for modeling, analyzing, and designing sensors and motions based on gaps. Information about gaps is used in a variety of existing work, perhaps most importantly in the Gap Navigation Tree (GNT). Though the GNT is astonishingly elegant as an algorithm, there remains scope to revisit and increase the precision with which the GNT's underlying assumptions are expressed, including —crucially— the definition of the sensing model it employs. Our framework brings improved rigor and exactitude to the existing theoretical results, and allows better understanding of the sensing capabilities demanded, which is valuable for practitioners attempting an implementation of the method. Specifically, we introduce *temporal relations*—a new form of perceptual information derived from signal-space continuities—and demonstrate their indispensability for gap navigation. We then propose a novel sensor and a new algorithm for the shortest path navigation problem with a learning phase. While the GNT addresses a similar problem, we prove that the GNT's sensor is incomparable to ours. By distilling the information in common between the GNT's sensor and our own, we obtain a new synthetic sensor resulting from the extraction or identification of essential information for gap navigation. Further, our analysis shows that the GNT's sensor can fail to recover this information in certain scenarios, whereas the new sensors naturally avoid this limitation, indicating improved robustness.

1 Introduction

Minimalist robotics [1–3,21,22] focuses on designing robot systems using limited sensing, action, and computation capabilities to complete specific tasks. As a line of work, it is valuable not only for guiding the development of low-cost robotic systems but also for establishing formalisms that describe both task requirements and robot capabilities. Notable past work has shown that much may be achieved with startlingly little.

Among various navigation strategies, the Gap Navigation Tree (GNT) algorithm [22] stands out for enabling a mobile robot to follow a distance-optimal path to a goal in an unknown planar environment without the robot being able to sense distances. In fact, it succeeds in the absence of all metric sensing of positions, distances, or angles. The GNT only needs gaps—discontinuities of depth around the robot, requiring the robot to detect the presence of gaps and their angular ordering.

Such gap sensors are interesting as a technology-agnostic mathematical abstraction because their outputs encode very succinct descriptions of regions of occlusion from the perspective of the robot. When visibility changes are triggered by the sensor undergoing motion, their intrinsically combinatorial representation of the environment is especially elegant. Yet, such sensors may also be conveniently implemented with different classes of available physical hardware, such as scanning LIDARs or cameras.

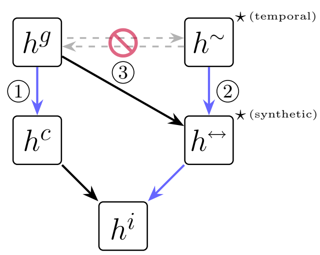


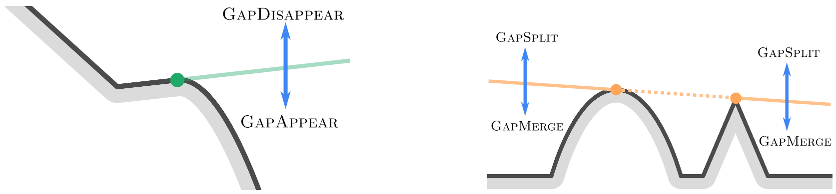
Fig. 1: Summary of new insights: A hierarchical structure illuminating the interrelationships between gap sensors studied in this paper, where arrows denote refinement [10]. The stars mark two new theoretical sensors introduced in this paper. The blue arrows specify the additional fact that two sensors differ only in terms of temporal relation information—a distinction this paper draws out. Arrow ① shows that sensor h^g used in the GNT algorithm enhances the cyclic order gap sensor h^c on the basis of temporal information. Arrows ② and ③ indicate that $h^\leftrightarrow$ is a sensor mutually subordinate to both h^g and $h^\sim$. The dashed arrows in gray with a slashed circle emphasizes the *absence* of any refinement relation (Lemma 2), indicating the incomparability between GNT's sensor h^g and our sensor $h^\sim$. Prior navigation employed h^g, but the new algorithms introduced here use the $h^\sim$ (incomparable) and $h^\leftrightarrow$ (weaker) sensors.

This work is concerned with better understanding the family of gap sensors as, beyond dovetailing nicely with the ambitions of minimalism, they involve considerable subtlety. Our primary contribution has been to scrutinize the phenomenon of gap tracking, where both knowledge of persistence and knowledge of changes in gaps is employed. Fig. 1 charts the relationship between different gap sensors, including new sensors we introduce, showing that these new variants also support navigation *à la GNT*. More importantly, these new sensor models help clarify how the GNT uses cyclic ordering data (answer: to establish an inter-temporal relation) and we identify constraints needed for this to be feasible, giving a concrete example in which it is violated.

1.1 Related Work

Visibility-based sensing and planning [7, 16, 19, 20], of which gap-based navigation is a particular sub-class, have been widely applied in the field of robot navigation, for instance, in so-called bug algorithms (e.g., [8] uses gaps and distances). Algorithms using gap sensing appear in such diverse work as [11, 13, 23]. We have already mentioned the original GNT algorithm, but a series of follow-ups include less idealized models in [12] and [14]. Extensions involving maintaining additional auxiliary data for multiply connected environments have been proposed (e.g., [15] is one example). The ideas underlying the GNT have also been adapted to handle pursuit-evasion problems [18].

We examine the interrelationships between sensors through LaValle's sensor lattice theory [10]. In this framework, a partial order is introduced to compare the capabilities of sensors, with a hierarchy of gap sensors presented as an illustrative example. Our results complement his analysis of gap sensors. The use of partial orders to compare robot systems also appears in [17] and [25], the latter extending sensor lattice theory by generalizing sensors into covers of their preimages.



(a) **(b)**

Fig. 2: An illustration summarizing gap events and their corresponding event lines. **(a)** The green dot marks an inflection point in the environment, and the green line represents an inflection line. Crossing this line from above causes an occluded region behind the line, producing a GapAppear event; crossing in the opposite direction causes a GapDisappear. **(b)** The dashed orange line is a bitangent line with tangent points marked by two orange dots, and the solid orange lines are the corresponding bitangent complement lines. Crossing either solid line from above causes a GapMerge wherein two occluded regions coalesce; crossing from below causes a GapSplit.

2 Preliminaries

We study the problem of robot navigation in a bounded, closed, and path-connected planar environment $\mathcal{E} \subset \mathbb{R}^2$. Following the standard convention (e.g., [22]), we invoke the general position assumption: no three points on the boundary $\partial\mathcal{E}$ are tangent to the same line segment $\ell \subset \mathcal{E}$. (For a polygonal edge on $\partial\mathcal{E}$ contained within ℓ, tangencies are defined at vertices rather than edge interiors.) This assumption rules out structurally unstable degeneracies in gap sensing. The robot's state space in environment $\mathcal{E}$ is $\mathcal{E} \times \mathbb{S}^1$, where $\mathbb{S}^1$ encodes orientation.

2.1 Gaps as Depth Discontinuities

For robot at state $x = (q, \varphi) \in \mathcal{E} \times \mathbb{S}^1$, ranging depth is captured by function $\rho : \mathbb{S}^1 \to \mathbb{R}_+$ where $\rho(\theta)$ is the distance from q to the environment boundary along the ray directed with angle θ relative to the robot's heading φ. A *gap* is defined as a discontinuity in ρ:

Definition 1 (Gap). *Define $\overset{\curvearrowright}{\rho}(\theta) := \lim_{\eta \curvearrowright \theta} \rho(\eta)$, $\overset{\curvearrowleft}{\rho}(\theta) := \lim_{\eta \curvearrowleft \theta} \rho(\eta)$ where $\lim_{\eta \curvearrowright \theta} / \lim_{\eta \curvearrowleft \theta}$ denote taking the limit in $\mathbb{S}^1$ from the increasing direction and the decreasing direction, respectively, under some parameterization. A gap is said to occur at direction θ if and only if $\overset{\curvearrowright}{\rho}(\theta) \neq \overset{\curvearrowleft}{\rho}(\theta)$.*

The gap triple $(\theta, \overset{\curvearrowright}{\rho}(\theta), \overset{\curvearrowleft}{\rho}(\theta))$ captures essential details of a gap at θ. We denote the set of all triples for the gaps detected at state x as $\mathcal{G}(x)$. To aid subsequent discussion, for a gap triple $g = (\theta, \overset{\curvearrowright}{\rho}(\theta), \overset{\curvearrowleft}{\rho}(\theta))$, let $\pi_\theta(g) := \theta$ and $\pi_{\text{near}}(g) := \min(\overset{\curvearrowright}{\rho}(\theta), \overset{\curvearrowleft}{\rho}(\theta))$. With only minor abuse of notation, we also map π_θ over the set of triples: for $\mathcal{G}(x) = \{g_1, g_2, \ldots, g_n\}$, with each $g_i = (\theta_i, \overset{\curvearrowright}{\rho}(\theta_i), \overset{\curvearrowleft}{\rho}(\theta_i))$, we define $\pi_\theta \circ \mathcal{G}(x) := \{\theta_1, \theta_2, \ldots, \theta_n\}$. Additionally, since $\pi_\theta \circ \mathcal{G}(x)$ must be a set of unique angles, the inverse function π_θ^{-1} is well-defined, mapping any θ_i back to g_i.

2.2 Gap Events

Gaps admit a visibility-based interpretation: each gap indicates the boundary of an invisible region. As the robot moves, the number of gaps changes through occurrences that we call *gap events*. Each gap event represents a fundamental transition in the visibility structure and thus conveys critical information about the environment's geometry.

Gap events fall into four distinct types: GAPAPPEAR, GAPDISAPPEAR, GAPSPLIT, and GAPMERGE. The first two correspond to the appearance and disappearance of a gap, occurring when the robot crosses an *inflection line*—a line extended along the slope of the boundary from an inflection point. Crossing the line one way gives a GAPAPPEAR, crossing the other produces a GAPDISAPPEAR. This is shown visually in Fig. 2a. The remaining two events involve interactions between gaps. In a GAPSPLIT, one gap divides in two; in a GAPMERGE, two gaps combine into one. These occur when the robot crosses a *bitangent complement line*, defined as the free extension of a bitangent line, with the crossing side again determining the event type (Fig. 2b).

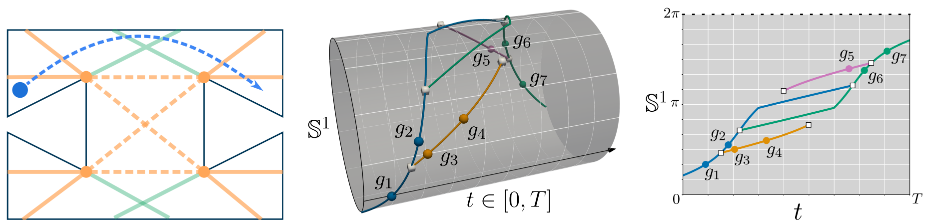
3 Coherence Across Time: Relations

Multiple instances of prior work (e.g., [22] and [18]) use essentially identical gap sensors that have high-level descriptions in natural language. It is worth being exact about what information a sensor provides because, as will be seen shortly, important nuance depends on assumptions that have been only tacit till now. We lay specific emphasis on providing a sufficiently rigorous treatment of the capabilities necessary to enable tracking of gaps. Models of different tracking abilities are claims about consistency across time. Many practical, data-sheet–like descriptions of sensors are statements about the output of a device for a single point in time. Either these sensor descriptions, strictly speaking, provide inadequate information to offer any tracking at all, or the sensor's signals are to be understood as having enough inter-timestep coherence so that matching is possible (under some extra, usually unstated assumptions). Neither is satisfactory—what is demanded is an explicit characterization of the postulated behavior of the sensor.

Shortly we will formalize the evolution of gaps over time. Continuity of motion results in various temporal relations between gaps, for which the overarching term 'coherence' will be used. This coherence captures the concept of *gap tracking* appearing within the literature and leads toward the exploration of new tracking capabilities.

3.1 Evolution of Gaps

As a robot traverses a trajectory $\sigma : [0,T] \to \mathcal{E} \times \mathbb{S}^1, T > 0$, the triple representations of gaps $\mathcal{G}(x)$ will change. This is captured by the composite function $\mathcal{G}_\sigma := \mathcal{G} \circ \sigma$, with angles extracted via $\Theta_\sigma := \pi_\theta \circ \mathcal{G}_\sigma$. The set-valued function Θ_σ can be visualized easily as curves on the $[0,T] \times \mathbb{S}^1$ cylinder, see Fig. 3 for example. These curves may contain *singular points* where one curve splits into two curves, or two curves merge into one. Mathematically, a pair $(\theta, \tau) \in \mathbb{S}^1 \times [0,T]$ is a singular point of Θ_σ if and only if there exist two continuous functions $\Omega_1, \Omega_2 : [t_a, t_b] \to \mathbb{S}^1$ such that $\Omega_1(t), \Omega_2(t) \in \Theta_\sigma(t)$, $\Omega_1(\tau) = \Omega_2(\tau) = \theta$, with $0 \le t_a < \tau < t_b \le T$, meeting either of the following two mutually exclusive conditions: 1) $\Omega_1(t) = \Omega_2(t)\ \forall t \in [t_a, \tau]$ and $\Omega_1(t) \neq \Omega_2(t)\ \forall t \in (\tau, t_b]$, 2) $\Omega_1(t) \neq \Omega_2(t)\ \forall t \in [t_a, \tau)$ and $\Omega_1(t) = \Omega_2(t)\ \forall t \in [\tau, t_b]$, where the former condition corresponds to a GAPSPLIT and the latter to a GAPMERGE.



(a) **(b)** **(c)**

Fig. 3: Visualized set-valued function Θ_σ and gap coherence relations. (a) An example environment with inflection, bitangent and bitangent complement lines shown. The blue dashed curve represents the robot's trajectory σ. **(b)** The robot's experience Θ_σ, for trajectory σ, visualized on the $[0,T] \times \mathbb{S}^1$ cylinder. **(c)** The cylinder and the Θ_σ function unwrapped to the plane. White squares mark the points on the curve that correspond to gap events: the singular points where two curves meet correspond to GapSplit and GapMerge events; the endpoints with $t \in (0,T)$ correspond to GapAppear and GapDisappear events. In this example, we have g_1 coheres with ($\rightsquigarrow$) gaps g_2, g_3, g_4, g_6, and g_7, $g_3 \rightsquigarrow\!\!\!\!\!\rightsquigarrow g_4$, $g_1 \overset{\mathrm{S}}{\rightsquigarrow} g_2$, $g_1 \underset{\mathrm{S}}{\rightsquigarrow} g_3$, $g_6 \overset{\mathrm{M}}{\rightsquigarrow} g_7$, and $g_5 \underset{\mathrm{M}}{\rightsquigarrow} g_7$.

3.2 Gap Coherence

Formally, gap coherence is a particular binary relation between gaps detected at different times:

Definition 2 (Gap coherence). *Let g_1 and g_2 be two gaps detected along σ at times t_1 and $t_2 \in [0,T]$ in directions θ_1 and θ_2 with $t_1 \le t_2$. We say g_1* coheres with *g_2, denoted as $g_1 \rightsquigarrow g_2$ if and only if there exists a continuous function $\Omega : [t_1,t_2] \to \mathbb{S}^1$ that satisfies $\Omega(t) \in \Theta_\sigma(t)\ \forall t \in [t_1,t_2]$ and $\Omega(t_1) = \theta_1$ and $\Omega(t_2) = \theta_2$.*

The coherence relation contains the data from which to derive an event's type and the specific gaps involved. Let $[t_1,t_2] \subseteq [0,T]$ be a time interval. At time t_2, if there exists a gap g such that no gap coheres with g at any times during $[t_1,t_2)$, then a GapAppear occurs at t_2 in which g appears. If two gaps g_a and g_b are observed at t_2 such that, for each time t in $[t_1,t_2)$, there exists one gap that coheres with them both, then a GapSplit occurs at t_2 resulting in g_a and g_b. In a similar way, we can identify a GapDisappear or a GapMerge with the relevant gaps. It will be useful to refine the relation $\rightsquigarrow$ to be more discriminating about the types of gap events. Also, for coherence across GapSplit or GapMerge events (e.g., $g_a \rightsquigarrow g_b$ and $g_a \rightsquigarrow g_c$ in the previous example), these may be further classified by distinguishing the continuity of the gap that represents the fore- from background. This yields the following.

Definition 3 (Event augmented coherence). *Let g_1 and g_2 be the two gaps detected at times t_1 and t_2 such that $g_1 \rightsquigarrow g_2$, and $\Omega(t)$ be the function connecting their angles θ_1 and θ_2 introduced in Definition 2. If for any $t \in (t_1,t_2)$, $(\Omega(t),t)$ is not a singular point, we denote $g_1 \rightsquigarrow\!\!\!\!\!\rightsquigarrow g_2$; if there exists one and only one singular point $(\Omega(t),t)$, we introduce the following notation:*

1. *In the case that $(\Omega(t),t)$ corresponds to a* GapSplit*, we denote $g_1 \overset{\mathrm{S}}{\rightsquigarrow} g_2$ if $\pi_{\mathrm{near}} \circ \pi_\theta^{-1} \circ \Omega$ is continuous at t, and $g_1 \underset{\mathrm{S}}{\rightsquigarrow} g_2$ otherwise.*

2. *In the case that* $(\Omega(t),t)$ *corresponds to a* GAPMERGE*, we denote* $g_1 \overset{\text{M}}{\rightsquigarrow} g_2$ *if* $\pi_{\text{near}} \circ \pi_\theta^{-1} \circ \Omega$ *is continuous at t, and* $g_1 \underset{\text{M}}{\rightsquigarrow} g_2$ *otherwise.*

Let $S^{\sim} := \{\rightsquigarrow\!\!\!\!\rightarrow, \overset{\text{M}}{\rightsquigarrow}, \underset{\text{M}}{\rightsquigarrow}, \overset{\text{S}}{\rightsquigarrow}, \underset{\text{S}}{\rightsquigarrow}\}$ *be a shorthand to denote this collection of* event-augmented coherence relations*.*

Definition 3 differentiates between gaps that cohere but for which there is no intervening gap event (with $\rightsquigarrow\!\!\!\!\rightarrow$) versus when some gap event is involved (with the other four). The latter are separated into $\overset{\text{S}}{\rightsquigarrow}$ and $\underset{\text{S}}{\rightsquigarrow}$ for GAPSPLITs, and into $\overset{\text{M}}{\rightsquigarrow}$ and $\underset{\text{M}}{\rightsquigarrow}$ for GAPMERGEs. Both gap coherence relations and event augmented coherence relations can be directly identified by visual inspection of Fig. 3.

We prove the following lemma on the $\rightsquigarrow\!\!\!\!\rightarrow$ relation.

Lemma 1. *Let g be a gap observed at time t. For any time $t' \geq t$, there exists at most one gap g' that satisfies $g \rightsquigarrow\!\!\!\!\rightarrow g'$.*

Proof. Consider the case that there exists such a gap g' at time t', then relation $g \rightsquigarrow\!\!\!\!\rightarrow g'$ implies that while g evolves into g', no gap event can occur that will impinge upon a gap along this evolution. This prohibits the corresponding $\Omega(t)$ from having singular points, thus guaranteeing no other gap exists at t' that satisfies the condition. □

These five event augmented relations are mutually exclusive and are exhaustive with respect to coherence, that is, each $\rightsquigarrow$ can be decomposed into a series of these relations. Relationships spanning longer periods of time can be constructed, but, for now, we hold off introducing our final temporal relation (in Definition 4).

4 Gap-chasing Motion Primitives

Although this paper's focus is primarily on sensing, action is allied with sensing in important ways. In navigation, sensing provides the information required to decide which action to execute; for gap navigation, the actions are defined as time-extended motion primitives that *chase gaps*. As continuous trajectories, these primitives condense action selection into operations that need only occur at specific discrete moments, namely gap events. Moreover, they constitute useful modularity that helps elevate decision making through the abstraction of low-level, implementation practicalities (such as individual motor control and information processing).

A gap-chasing motion primitive is defined for each gap currently observed by a robot. If a *chase-gap-g* motion is activated at time t, the robot starts moving in the direction of g. Since the direction of the gaps may vary during execution, the robot must track and follow the target gap through those changes. Mathematically this is represented via the $\rightsquigarrow\!\!\!\!\rightarrow$ relation: subsequent to t, if there exists a gap g' such that $g \rightsquigarrow\!\!\!\!\rightarrow g'$, the robot moves toward g'. If there is no such g', the motion primitive completes and the robot stops. Lemma 1 establishes that this leads to a well-defined behavior.

The trajectory induced by gap chasing consists of two phases. First, if the robot is within the environment's interior, it approaches the boundary along a tangent line. Having gained the boundary, the robot transitions to wall-following because the gap must now be observed in a direction tangent to the boundary. Wall following continues until the gap being chased undergoes a GAPSPLIT or GAPDISAPPEAR, at which moment the robot decides on the next gap to chase, or instead to halt. Such a trajectory will lie on the environment's reduced visibility graph [9] which encodes all the shortest paths.

5 Gap-sensing Based on Coherence

Shifting perspective to consider the data required to decide which motion primitive to execute, we adopt the notion of *abstract sensors* [6] wherein sensors are defined in terms of the information they provide. The implications for specific implementation and the relationship to perception more generally will be discussed at the end of the section.

5.1 Gap Sensing Principles and Baseline Sensors

A remarkable feature of gap navigation is that planning requires surprisingly little information. The sensor is consulted only at the initial time and at subsequent gap events, and its output carries no quantitative data such as distances or angles. Between events, the robot executes gap-chasing motion primitives, and those behaviors entirely encapsulate all metric aspects that are relevant. Despite this apparent simplicity, subtle choices in sensor design have significant consequences for navigation capabilities.

We start with bare essentials. First off, we demand sensor feedback contain a distinct indicator for each currently observed gap. An indicator provides no information other than the existence of the gap it corresponds to. They form the foundational signal that more sophisticated sensors augment.

Second, a gap sensor is required to output a signal at the initial time and upon each gap event. Let σ be a robot trajectory starting at $t_0 = 0$, and let $\{t_1, t_2, \ldots, t_n\}$ be the times at which gap events occur along σ. Then, a gap sensor must provide an output at each t_i, $i \in \{0, 1, \ldots, n\}$. This ensures that details about each gap event can be obtained by comparing the sensor signal before and at the event.

These two requirements and only these two are captured in the baseline sensor:

Sensor 0 (Gap indicator sensor)**.** *A gap indicator sensor h^i outputs a set-valued signal $\mathcal{I}_i$ at t_i, where each gap g in $\mathcal{G}(\sigma(t_i))$ corresponds one-to-one to an indicator $\iota \in \mathcal{I}_i$.*

The gap indicator sensor h^i is a weak sensor: it cannot distinguish GAPAPPEAR *vs* GAPSPLIT, or GAPDISAPPEAR *vs* GAPMERGE, nor can it identify which indicators participate in an event. The reason is that the indicators are *existential* and *instantaneous*—each $\iota_j \in \mathcal{I}_i$ corresponds only to the gap detected at t_i, and is *irrelevant* to gaps observed at other times. With only indicators $\mathcal{I}_{i-1}$ and $\mathcal{I}_i$, the change of cardinality of the gap sets can be inferred, but not the way gaps at t_{i-1} evolve into those at t_i, i.e., through a GAPAPPEAR or GAPSPLIT. To our knowledge, no algorithm has been proposed that successfully navigates using only this.

An extension could include suitable additional information; the cyclic ordering is an example, having been put to good use in [22]:

Sensor 1 (Cyclic order gap sensor)**.** *Cyclic order gap sensor h^c extends h^i by outputting a signal $s_i = (\mathcal{I}_i, C_i)$ at t_i where $C_i = \{(\iota_a, \iota_b) \in \mathcal{I}_i \times \mathcal{I}_a \mid g_b$ is the clockwise neighbor of $g_k\}$, with ι_a and ι_b as the indicators of gap g_a and g_b respectively.*

Compared to h^i, the gap sensor h^c provides the relative angular information of gaps. At first blush, one might presume that the cyclic order reveals useful gap details, however h^c does no better than h^i with respect to events. Both sensors output $\mathcal{I}_i$ in which gap indicators are of equal standing: each ι_i signals only the presence of a gap and carries no distinguishing information. The addition of cyclic ordering information does not break the equivalence—every single gap is assigned a clockwise and an anti-clockwise neighbor, but none is privileged in a way that discloses its information across time. Consequently, h^c is no more acute than h^i in discriminating gap events.

Different gap event types have distinct visibility implications, altering choices during navigation; similarly, no algorithm is known that succeeds with h^c. (This ought to surprise readers familiar with the GNT!) Gap indicators are too weak to be gap identifiers: the next step is to have sensors capturing coherence of gaps across events.

5.2 Temporal-relation Sensors

In contrast to conventional sensors that report only *instantaneous information* determined by the current environment, temporal-relation sensors produce signals that report relationships describing the evolution of events. As an example, we extend the cyclic order gap sensor to the GNT sensor. In addition to the cyclic order, the GNT sensor provides "$\rightsquigarrow$" relations.

Sensor 2 (GNT sensor)**.** *A GNT sensor h^g extends h^c by outputting a signal $s_i = (\mathcal{I}_i, C_i, R_i^{\rightsquigarrow})$ at t_i where $R_0^{\rightsquigarrow} = \varnothing$ and $R_{i>0}^{\rightsquigarrow} = \{(\iota_a, \iota_b) \in \mathcal{I}_{i-1} \times \mathcal{I}_i \mid g_a \rightsquigarrow g_b\}$, where ι_a and ι_b are the indicators of gap g_a and g_b respectively.*

This sensor is equivalent to that used in [22]. There, gaps are labeled by gap identifiers and a list of identifiers is maintained. Across a gap event: identifiers can be added to, be dropped from, or stay within the list, also they may be combined to create a new composite gap label. We make explicit the information these identifiers carry through temporal relations: adding and dropping identifiers signifies the absence of a coherence relation, staying in the list indicates a one-to-one $\rightsquigarrow$ relation between two gaps, and composite labels indicate two-to-one $\rightsquigarrow$ relations.

From this vantage point, we now propose a novel sensor (the first of two) that reports stronger temporal relations.

Sensor 3 (Event-augmented coherence sensor)**.** *An event-augmented coherence sensor $h^{\sim}$ extends h^i by outputting a signal $s_i = \left(\mathcal{I}_i, \left\{R_i^{\sim} \mid \sim \in S^{\sim}\right\}\right)$ at t_i where $R_0^{\sim} = \varnothing$ and $R_{i>0}^{\sim} = \{(\iota_a, \iota_b) \in \mathcal{I}_{i-1} \times \mathcal{I}_i \mid g_a \sim g_b\}$, for each $\sim \in S^{\sim}$.*

5.3 Inter-relationships between Gap Sensors

Sensor lattice theory [10] provides one way to investigate the relative capabilities of distinct sensors. Treating a sensor as a map from states to output signals, it considers how the state space is divided into subsets that produce identical sensor outputs: any such map induces a partition of the state space through its preimages. The notion of partition refinement then provides partial order (denoted $\succeq$) on sensing capabilities.

Next, we show that $h^{\sim}$ and h^g are incomparable. (Later, §7.2 returns to the topic, completing the lattice.)

Lemma 2. *$h^{\sim} \not\succeq h^g$ and $h^g \not\succeq h^{\sim}$.*

Proof. Degeneracies apart, $h^{\sim}$ has no basis on which to simulate the cyclic order provided by h^g, therefore $h^{\sim} \not\succeq h^g$. For h^g, although it can recognize GAPSPLIT and GAPMERGE events, it cannot distinguish $\overset{S}{\rightsquigarrow}$ *vs* $\underset{S}{\rightsquigarrow}$ and $\overset{M}{\rightsquigarrow}$ *vs* $\underset{M}{\rightsquigarrow}$. □

Both $h^{\sim}$ and h^{g} may be implemented with a range sensor. The five relations in $S^{\sim}$ are distinguished by using $\rightsquigarrow$-based gap tracking and then using a distance ranking for the pair of gaps which merge in a GAPMERGE or split from a GAPSPLIT. The GNT sensor h^{g} also requires $\rightsquigarrow$-based gap tracking, with cyclic order of gaps, additionally. As any gap sensor must scan the distance at all azimuth angles, it is not an extravagant requirement for the sensor to provide either form of the extra information.

5.4 Concern: Are Temporal Relations Sensable and Sensible?

In the preceding, by directly modeling temporal interrelationships, we uncover a dimension of sensing information that is ‘orthogonal’ to the usual instantaneous information; doing so has furnished a richer family of abstract sensors to be considered for gap navigation. This perspective clarifies what information is fundamentally required: not numerical data such as distances or angles, but combinatorial relations between gaps across events. From an implementation standpoint, computing the temporal relations is unavoidable. For instance, in [14] the $\rightsquigarrow$ relation is recovered through a data-association algorithm. Making these relations explicit provides guidance on which instantaneous measurements, together with suitable assumptions, suffice to reconstruct the required temporal information. Later, we show how the combination of sensor outputs and motion assumptions can successfully yield the temporal relations essential for gap navigation, and how the absence of such assumptions can cause navigation algorithms to fail.

One possible doubt is whether temporal relations should be considered sensor feedback, since they are not commonly available from physical devices directly. In fact, similar information has already been treated as sensing in the literature. For instance, [10] models time-extended trajectories as sensor inputs, naturally encompassing temporal information derived from trajectories. Moreover, the relational sensors we describe can be seen as related to event-based sensors [24], in which the sensor maintains an internal state and outputs are triggered upon changes. Also, high-level perceptual information is frequently regarded as feedback from virtual sensors [5, 10]. For example, [5, Appendix B] discusses a “grandmother sensor” which returns a binary bit indicating whether a grandmother exists in the field of view. In that vein, temporal relations align with established notions of sensing, and it is consistent to classify them as such.

6 Optimal Navigation after a Learning Phase

To demonstrate the preceding treatment’s power, we now apply the sensors defined in the previous section to a shortest path navigation problem, as studied within the existing literature.[1]

6.1 Problem Definition

To grant the robot the capability to reach a goal position, the concept of a *landmark* is needed—each is a specific position in the environment $\mathcal{E}$. Every landmark has a unique label ℓ; if the landmark is visible from the gap sensor, the label ℓ will show in the

[1] Technically, this is a moderate generalization of that in [22], see §8.

sensor's output $\mathcal{I}_i$. In this case, a motion primitive is available to chase ℓ, which causes the robot to move straight to ℓ and stop at ℓ. To solve a navigation problem, the robot must perform a sequence of gap-chasing actions until some goal landmark is reached.

Next, we define the shortest path navigation problem that we consider; it involves navigation in an unknown environment. The robot first traverses a learning path λ and then uses the sensing data collected along λ to navigate to the goal position.

Question 1 (Optimal navigation query after a learning phase). An optimal navigation query after a learning phase, $\mathcal{Q}(\lambda, \ell)$, is defined as follows: Let $\mathcal{E}$ be an environment with a landmark ℓ and $\lambda : [0,T] \to \mathcal{E} \times \mathbb{S}^1$ be a robot trajectory such that ℓ is visible from some state $\lambda(t)$. Without prior knowledge of $\mathcal{E}$, a robot with sensor h traverses λ and collects the sensing history $h(\lambda)$. The robot must navigate from $\lambda(T)$ to ℓ through a shortest path via a sequence of gap-chasing actions, computed using only $h(\lambda)$ and ℓ.

Algorithm 1: Gap Navigation with a $h^{\sim}$ Sensor

Input: sensing history $h^{\sim}(\lambda) = \{s_1, s_2, \ldots, s_n\}$, label ℓ
/* ·················· Process Learning Phase Data ················ */
$\iota^* \leftarrow \ell$; $A \leftarrow$ empty stack
for $i \leftarrow 1$ **to** n **do**
    $(\mathcal{I}, \{R^{\sim} \mid \sim \in S^{\sim}\}) \leftarrow s_i$ // Unpack signal
    $\iota^*, A \leftarrow$ UPDATE-A $(\iota^*, A, \mathcal{I}, \{R^{\sim} \mid \sim \in S^{\sim}\})$
/* ························· Execution Phase ······················ */
while ℓ **not** *reached* **do**
    Chase the gap indicated by ι^*
    $(\mathcal{I}, \{R^{\sim} \mid \sim \in S^{\sim}\}) \leftarrow$ latest observation from $h^{\sim}$
    $\iota^*, A \leftarrow$ UPDATE-A $(\iota^*, A, \mathcal{I}, \{R^{\sim} \mid \sim \in S^{\sim}\})$

function UPDATE-A $(\iota^*, A, \mathcal{I}, \{R^{\sim} \mid \sim \in S^{\sim}\})$
    if $\exists \iota \in \mathcal{I}, (\iota^*, \iota) \in R^{\rightsquigarrow}$ **then**
        $\iota^* \leftarrow \iota$
    else if $\exists \iota \in \mathcal{I}, (\iota^*, \iota) \in R^{\overset{\mathrm{M}}{\curvearrowright}}$ **then**
        A.push($\overset{\mathrm{M}}{\curvearrowright}$); $\iota^* \leftarrow \iota$
    else if $\exists \iota \in \mathcal{I}, (\iota^*, \iota) \in R^{\underset{\mathrm{M}}{\curvearrowleft}}$ **then**
        A.push($\underset{\mathrm{M}}{\curvearrowleft}$); $\iota^* \leftarrow \iota$
    else if A.top() $= \overset{\mathrm{M}}{\curvearrowright}$ **and** $\exists \iota \in \mathcal{I}, (\iota^*, \iota) \in R^{\overset{\mathrm{S}}{\curvearrowright}}$ **then**
        A.pop(); $\iota^* \leftarrow \iota$
    else if A.top() $= \underset{\mathrm{M}}{\curvearrowleft}$ **and** $\exists \iota \in \mathcal{I}, (\iota^*, \iota) \in R^{\underset{\mathrm{S}}{\curvearrowleft}}$ **then**
        A.pop(); $\iota^* \leftarrow \iota$
    return ι^*, A

Lemma 3. *A robot with a $h^{\sim}$ sensor solves any $\mathcal{Q}(\lambda,\ell)$ by Algorithm 1.*

Proof (sketch): At the end point of the learning path $\lambda(T)$, if ℓ is visible, then directly chasing ℓ solves the query; if ℓ is not visible, it must have merged into a sequence of gaps. A robot chasing these gaps in the reversed order of merging will experience the inversion—a sequence of GAPSPLITs which eventually exposes ℓ after traversing a shortest path to ℓ. The difficulty in undertaking this process is that after the gap being chased splits, the robot must decide which gap to follow from the two gaps that emerge from the GAPSPLIT.

This issue is solved by looking closely at the dual processes of GAPMERGE and GAPSPLIT. In a GAPMERGE, the nearer gap stays in the sensor's view, while the farther gap is occluded behind it. The temporal relations express this: the more distant gap merges through a $\underset{\text{M}}{\rightsquigarrow}$, the nearer through a $\overset{\text{M}}{\rightsquigarrow}$. Traversing in reverse gives a GAPSPLIT and what was previously the distant gap is still the farther one, splitting out through a $\underset{\text{S}}{\rightsquigarrow}$, while the nearer gap splits out through a $\overset{\text{S}}{\rightsquigarrow}$. This property allows the robot to select which gap to chase from between the newly split out gaps.

The algorithm utilizes this property by maintaining two variables, ι^*, the indicator of the current gap to chase, and A, a stack encoding how ℓ is obscured through a sequence of GAPMERGEs. It takes two inputs: the sensing history $h^{\sim}(\lambda) = \{s_1, s_2, \ldots, s_n\}$ and the landmark label ℓ. The stack A is constructed incrementally by processing each s_i (lines 1–4) using the UPDATE-A function declared at line 9. Initially, $\iota^* = \ell$ and A is an empty stack. If no event occurs on the gap indicated by ι^*, the next ι^* is identified using the $\rightsquigarrow\!\!\!\rightarrow$ relation. Whenever ι^* merges into another gap, UPDATE-A records whether it merged through a $\overset{\text{M}}{\rightsquigarrow}$ (line 13); or a $\underset{\text{M}}{\rightsquigarrow}$ (line 15). The indicator of the gap that ι^* merged into becomes the next ι^*. If ι^* undergoes a GAPSPLIT, UPDATE-A identifies the new ι^* from the two new gaps by matching their temporal relations to the element on the top of stack A: $\overset{\text{S}}{\rightsquigarrow}$ matches with $\overset{\text{M}}{\rightsquigarrow}$, while $\underset{\text{S}}{\rightsquigarrow}$ matches with $\underset{\text{M}}{\rightsquigarrow}$ (lines 16 and 18). The top of A is then removed.

After the entire history $h^{\sim}(\lambda)$ has been processed, the algorithm enters the execution phase. The robot chases the gap indicated by ι^* until it encounters a GAPSPLIT, at which point UPDATE-A is invoked again to modify A and identify the new ι^*. Navigation concludes once A is empty. △

7 A Synthetic Sensor

Lemma 3 indicates that sensor $h^{\sim}$ provides sufficient information to solve Question 1, i.e., any query $\mathcal{Q}(\lambda,\ell)$. Similarly, in [22], the GNT algorithm with sensor h^g yields optimal navigation after an initial phase of exploration. The two sensors, $h^{\sim}$ and h^g, output signals that are obviously distinct. The difference is not a question of a superficial transformation of data from one form into another because their lack of comparability, as expressed Lemma 2, means that each sensor has some underlying information which cannot be obtained from the other. Yet both $h^{\sim}$ and h^g are sufficiently strong to solve instances of $\mathcal{Q}(\lambda,\ell)$.

This leads to a suspicion that perhaps both $h^{\sim}$ and h^g are 'over-sensing' somehow. And, on the quest for minimality, then to ask: might one identify a weaker sensor, one that is *mutually subordinate* to both h^g and $h^{\sim}$, but which suffices for the navigation task? The answer is yes and we will provide such a sensor next. We call this a *synthetic* sensor because, rather than looking either at physical devices or the task itself, it was obtained by pondering how to distill common information from the two other sensors.

7.1 Gap Association: A Mutually Subordinate Sensor

As before, we start with a relation capturing associations between gaps observed along a robot trajectory σ. The following relation holds over longer duration than those in Definitions 2 and 3 because it continues to hold when multiple events intervene.

Definition 4 (Association). *Gap g_1 and g_2 are* associated*, written as $g_1 \leftrightarrow g_2$, if one of the following conditions holds:*

1. $\forall g,\ g \leftrightarrow g$;
2. $g_1 \leftrightarrow g_2 \implies g_2 \leftrightarrow g_1$;
3. $g_1 \leftrightarrow g_2,\ g_2 \leftrightarrow g_3 \implies g_1 \leftrightarrow g_3$;
4. $g_1 \leadsto\!\!\!\!\!\leadsto g_2 \implies g_1 \leftrightarrow g_2$;
5. $g_1 \overset{\text{M}}{\rightsquigarrow} g_a \leftrightarrow g_b \overset{\text{S}}{\rightsquigarrow} g_2 \implies g_1 \leftrightarrow g_2$;
6. $g_1 \underset{\text{M}}{\rightsquigarrow} g_a \leftrightarrow g_b \underset{\text{S}}{\rightsquigarrow} g_2 \implies g_1 \leftrightarrow g_2$.

The essence of this relation is in items 5) and 6), which express an intuitive notion of persistence across 'bracketed' merge and split pairs.

Next, we define an abstract association sensor:

Sensor 4 (Gap association sensor)**.** *A gap association sensor $h^{\leftrightarrow}$ extends h^i by outputting a signal $s_i = (\mathcal{I}_i, R_i^{\rightsquigarrow}, R_i^{\leftrightarrow})$ at t_i where $R_i^{\rightsquigarrow}$'s equal those from the GNT sensor, $R_0^{\leftrightarrow} = \varnothing$, and $R_{i>0}^{\leftrightarrow} = \{(\iota_a, \iota_b) \in \mathcal{I}_j \times \mathcal{I}_i \mid i > j, g_a \leftrightarrow g_b\}$.*

7.2 Lattice of Gap Sensors

With Sensors 0–4 defined, we now are in a position to finish the discussion begun briefly in §5.3. The lattice of these sensors is illustrated in Fig. 1. The figure contains five refinements, among which $h^g \succeq h^c$, $h^c \succeq h^i$ and $h^{\leftrightarrow} \succeq h^i$ are obvious, as the former sensors directly build upon the latter ones. The refinement $h^{\sim} \succeq h^{\leftrightarrow}$ is a result of Definition 4, which states that the $\leftrightarrow$ relation can be derived from $\leadsto\!\!\!\!\!\leadsto$, $\overset{\text{M}}{\rightsquigarrow}$, $\underset{\text{M}}{\rightsquigarrow}$, $\overset{\text{S}}{\rightsquigarrow}$ and $\underset{\text{S}}{\rightsquigarrow}$ relations. That $h^{\sim}$ and h^g are incomparable (Lemma 2) is indicated in Fig. 1 by the gray broken arrows and the forbidden symbol. A single arrow remains:

"Proposition" 1. $h^g \succeq h^{\leftrightarrow}$.

Proof (sketch): We must show that every $\leftrightarrow$ relation provided by a $h^{\leftrightarrow}$ can be derived from the information produced by a h^g sensor. The first three conditions of Definition 4 establish that $\leftrightarrow$ is an equivalence relation. Closure using those three properties allows one to populate pairs in the relation. What is required, then, is to obtain the other primitive elements of the $\leftrightarrow$ relation using only data obtained from h^g. The simplest is $\leadsto\!\!\!\!\!\leadsto$, in condition 4. From the h^g sensor's output, they can be detected as one-to-one $\rightsquigarrow$ relations. The more involved primitives are the cases in conditions 5 and 6:

Algorithm 2: Gap Navigation with a $h^{\leftrightarrow}$ Sensor

Input: sensing history $h^{\leftrightarrow}(\lambda) = \{s_1, s_2, \ldots, s_n\}$, label ℓ

/* ················ Process Learning Phase Data ················ */

1 $\iota^* \leftarrow \ell$; $B \leftarrow$ empty list; B.append(ℓ)

2 **for** $i \leftarrow 1$ **to** n **do**

3 $\quad (\mathcal{I}, R^{\rightsquigarrow}, R^{\leftrightarrow}) \leftarrow s_i$ // Unpack signal

4 $\quad \iota^*, B \leftarrow$ UPDATE-B $(\iota^*, B, \mathcal{I}, R^{\rightsquigarrow}, R^{\leftrightarrow})$

/* ················ Execution Phase ················ */

5 **while** ℓ **not** *reached* **do**

6 $\quad$ Chase the gap indicated by ι^*

7 $\quad (\mathcal{I}, R^{\rightsquigarrow}, R^{\leftrightarrow}) \leftarrow$ latest observation from $h^{\leftrightarrow}$

8 $\quad \iota^*, B \leftarrow$ UPDATE-B $(\iota^*, B, \mathcal{I}, R^{\rightsquigarrow}, R^{\leftrightarrow})$

9 **function** UPDATE-B $(\iota^*, B, \mathcal{I}, R^{\rightsquigarrow}, R^{\leftrightarrow})$

10 $\quad$ **if** $\exists \iota \in \mathcal{I}, \exists b_i \in B$ such that $(b_i, \iota) \in R^{\leftrightarrow}$ **then**

11 $\quad\quad B \leftarrow [b_1, b_2, \ldots, b_i]$; $\iota^* \leftarrow \iota$

12 $\quad$ **else if** $\exists \iota \in \mathcal{I}$ such that $(\iota^*, \iota) \in R^{\rightsquigarrow}$ **then**

13 $\quad\quad B$.append(ι); $\iota^* \leftarrow \iota$

14 $\quad$ **return** ι^*, B

$g_1 \overset{\text{M}}{\rightsquigarrow} g_a \mathrel{\substack{\rightsquigarrow\\\rightsquigarrow}} g_b \overset{\text{S}}{\rightsquigarrow} g_2$ and $g_1 \underset{\text{M}}{\rightsquigarrow} g_a \mathrel{\substack{\rightsquigarrow\\\rightsquigarrow}} g_b \underset{\text{S}}{\rightsquigarrow} g_2$. Note that h^g cannot distinguish $\overset{\text{M}}{\rightsquigarrow}$ from $\underset{\text{M}}{\rightsquigarrow}$, or $\overset{\text{S}}{\rightsquigarrow}$ *vs* $\underset{\text{S}}{\rightsquigarrow}$. Suppose nevertheless, $g_1 \overset{\text{M}}{\rightsquigarrow} g_a$ and $g_1' \underset{\text{M}}{\rightsquigarrow} g_a$, whereupon g_a evolves into g_b, and hence $g_b \overset{\text{S}}{\rightsquigarrow} g_2$ and $g_b \underset{\text{S}}{\rightsquigarrow} g_2'$. Then their relative cyclic ordering is consistent: g_1 is immediately (anti)clockwise of g_1' if and only if g_2 is immediately (anti)clockwise of g_2'. Thus $\leftrightarrow$ rules can be formed using disambiguating cyclic ordering (relative anti/ clockwise) data. Thereafter, all primitive relations can be derived using h^g, with the remaining generated through repeated application of 1, 2, and 3. $\triangle$

7.3 Optimal Navigation Redux

To confirm that $h^{\leftrightarrow}$ is sufficient for any experience-based optimal navigation query $\mathcal{Q}(\lambda, \ell)$, we provide an algorithm.

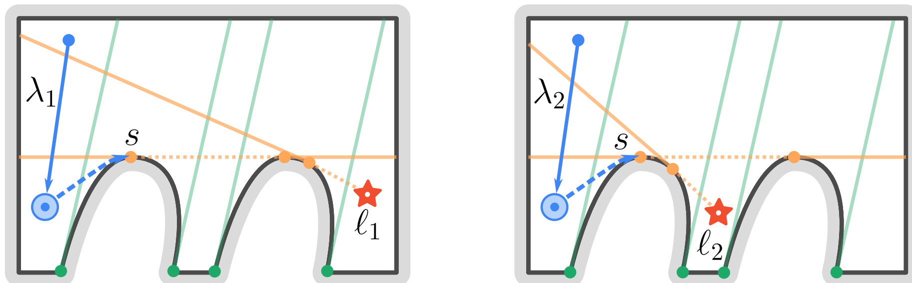
Lemma 4. *A robot with $h^{\leftrightarrow}$ sensing solves any $\mathcal{Q}(\lambda, \ell)$ via Algorithm 2.*

Proof (sketch): Algorithm 2 is structurally similar to Algorithm 1. The roles of ι^*, B, and UPDATE-B correspond to ι^*, A, and UPDATE-A, respectively, with the distinction that B stores indicators rather than coherence relations.

The UPDATE-B function is simpler than UPDATE-A since it needn't record the details of how a gap merges or splits. Using existence of a pair in $R^{\leftrightarrow}$ (line 10), the next ι^* can be directly identified as the definition of the $\leftrightarrow$ relation captures the processes at line 10, 16 and 18 of Algorithm 1. Whenever $R^{\leftrightarrow}$ contains no $\leftrightarrow$ relation to any b_i, then the gap indicated by ι^* undergoes a GAPMERGE. The indicator of the gap into which ι^* merged is identified using the basic $\rightsquigarrow$ relation (line 12) and appended to B. $\triangle$

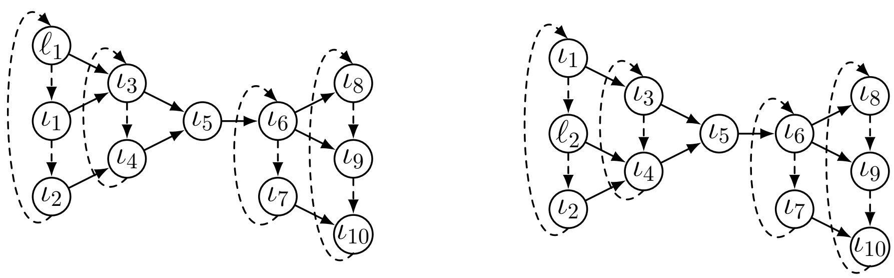
8 A Not-So-Counterexample

Next, we illustrate how temporal relations are valuable in aiding systematic thinking about the transformation of information. We carefully examine the sensing information



(a) Query $\mathcal{Q}(\lambda_1, \ell_1)$. **(b)** Query $\mathcal{Q}(\lambda_2, \ell_2)$.

Fig. 4: Distinct navigation queries with identical sensing histories. **(a)** Shows navigation query $\mathcal{Q}(\lambda_1, \ell_1)$. Event lines are illustrated as in Fig. 2b & 2a, the star denotes the landmark ℓ_1, the solid blue arrow is learning path λ_1, and the dashed blue arrow shows the trajectory of the first gap-chasing action. **(b)** Navigation query $\mathcal{Q}(\lambda_2, \ell_2)$ is similar, and is shown using same visualization conventions.



(a) History corresponding to Fig. 4a. **(b)** History corresponding to Fig. 4b.

Fig. 5: Sensing histories from a h^g sensor in different navigation queries. **(a)** The sensing history from a h^g sensor in query $\mathcal{Q}(\lambda_1, \ell_1)$ as the robot traverses λ_1 and then reaches s by gap-chasing motions. **(b)** The analogous history for $\mathcal{Q}(\lambda_2, \ell_2)$ as the robot traverses λ_2 and then reaches s. The combinatorial structures of the two sensing histories are isomorphic. Since all ιs are indicators, their subscripts serve merely as notational distinctions, and the two sensing histories are thus identical.

and the assumptions involved in the navigation problem; by adopting the attitude that the temporal relations have primacy, we ask how those are derived from other data.

The $\leftrightarrow$ relation is defined purely through temporal coherence relations. Sensor h^g uses spatial information, namely cyclic orders, to recover $\leftrightarrow$ in an indirect way. In this section we give an example constructed so that such a mechanism fails and a robot with h^g, no matter what algorithm it employs, must fail with some navigation queries $\mathcal{Q}(\lambda, \ell)$. That is to say, $h^g \succeq h^{\leftrightarrow}$ is not correct in general and that arrow ③ ought not to appear in Fig. 1! Nevertheless, the example does *not* prove that the GNT algorithm is incorrect either. That subtlety will be explained shortly.

Consider two navigation queries $\mathcal{Q}(\lambda_1, \ell_1)$ and $\mathcal{Q}(\lambda_2, \ell_2)$, illustrated in Figs. 4a and 4b, respectively. We claim:

Lemma 5. *There exists no algorithm that a robot with an h^g sensor can apply to solve $\mathcal{Q}(\lambda_1, \ell_1)$ and $\mathcal{Q}(\lambda_2, \ell_2)$ deterministically.*

Proof. For the robot to reach the respective goal position, in both cases, the first action at $\lambda_1(T)$ and $\lambda_2(T)$ is to chase the only gap in view. This results in the robot reaching position s. Up to this moment, the robot's sensing histories in the two environments are indistinguishable—as illustrated Figs. 5a & 5b are isomorphic, being identical up to (arbitrary) indicator labels. But the next gap to chase to reach the goal through a shortest path differs in $\mathcal{Q}(\lambda_1, \ell_1)$ *vs* $\mathcal{Q}(\lambda_2, \ell_2)$. □

The h^g sensor's defect becomes apparent at the second GAPMERGE along the learning paths λ_1/λ_2, where the only two gaps which are observable merge into one. The cyclic order, though helpful at disambiguating the three gaps before the first GAPMERGE, is inadequate to distinguish the two gaps before the second GAPMERGE as both gaps were the clockwise neighbor of the other. This inevitably leads to confusion in the action choice after the GAPSPLIT at s.

However, the GNT algorithm cunningly avoids the above issue. The GNT requires that, instead of being an arbitrary trajectory, λ also be generated by gap-chasing actions. The source of ambiguity is eliminated on this class of paths:

Lemma 6. *Along paths generated by gap-chasing actions, there must be at least three gaps before a* GAPMERGE *and at least two gaps before a* GAPSPLIT.

Proof. A GAPMERGE never involves the gap currently being chased. Thus, before a GAPMERGE, there must exist two gaps that are going to merge in addition to the gap being chased. For a GAPSPLIT, if it does not occur on the gap being chased, then there exist two gaps before the event. If it does, recall that the gap-chasing action commands the robot to first reach the environment boundary through a straight line and then follow the wall. On the straight line path, the gap cannot split, so the GAPSPLIT must occur when the robot is already on the boundary, where at least two gaps can be observed along the forward and backward directions of the wall. □

Lemma 7. *Let the robot traverse only trajectories generated by a sequence of gap-chasing actions; then,* $h^g \succeq h^{\leftrightarrow}$.

Proof. Lemma 6 ensures that h^g can, using the cyclic order, always distinguish between the two gaps merged in a GAPMERGE or the two gaps split out of a GAPSPLIT. This guarantees that the $\leftrightarrow$ relation can always be retrieved. □

In other words, with carefully chosen assumptions regarding the sensor and motions (and really their interplay), the essential temporal relation can be deduced. The explicit relations impose enough rigor on the analysis that this subtlety, which might easily be overlooked, is revealed. The upshot is that the arrow ③ in Fig. 1 is justified but only when considering a constrained set of experiences. Indeed, "Proposition" 1 bears quotation marks because it requires this additional proviso in order to hold. (Within the argument, the phrase 'disambiguating cyclic ordering' is the crux, because the ordering fails to disambiguate when gaps are simultaneously clockwise and anticlockwise neighbors.) This result bears a curious relationship to the work in [4] on the visibility graph reconstruction problem for polygonal environments; that work showed that being limited to moving on the boundary is a fatal impairment for that task. Here, the constraint is helpful for the GNT, despite there being substantial similarity in the high-level objectives of the robot.

Finally, this also leads one to recognize that the correctness of GNT is quite delicate in regards its idealized model: if the robot has a positive extent, or if gap-chasing motions deviate during execution, then ambiguity akin to that in Fig. 4 arises. Algorithms 1 & 2 may be considered more robust in that the specific sense they do not suffer this defect.

9 Conclusion

The present work provides a new, and we believe deeper, understanding of gap sensing and leads to improved navigation robustness. This paper employed an approach of systematically making explicit what was previously only implicit: we did this in representing temporal relationships that underpin gap tracking (formalized rigorously through binary relations) and expressed how those relational data are key to gap-chasing behaviors. Putting primacy on temporal relationships, we examined how specific sensors in the literature are, essentially, establishing those relational associations. Further, careful analysis of gap sensor dominance led to a search to identify the informational 'common denominator', which yielded a novel, synthetic sensor. One hesitates, of course, to draw too strong a parallel, but the sense that the theory suggested there "ought to be" some sensor which had not yet been identified has a bit of the flavor of Le Verrier and Adams's calculations suggesting where to look for what we now know of as Neptune.

Direct modeling of temporal interrelationships to describe information which is, as a manner of speaking, orthogonal to the instantaneous data, reveals a dimension of sensing that was hidden previously—somehow wrapped up in assumptions of data association. This has been a fruitful approach for gap navigation as it enabled the introduction of an alternative to the GNT algorithm using a new weaker sensor, yet which can solve the problem of optimal navigation under a wider set of input conditions.

References

1. Agarwal, P., Collins, A., Harer, J.: Minimal trap design. In: IEEE ICRA. vol. 3, pp. 2243–2248 vol.3 (2001). https://doi.org/10.1109/ROBOT.2001.932956
2. Alam, T., Bobadilla, L., Shell, D.A.: Minimalist robot navigation and coverage using a dynamical system approach. In: 2017 First IEEE International Conference on Robotic Computing (IRC). pp. 249–256 (2017). https://doi.org/10.1109/IRC.2017.23
3. Connell, J.H.: Minimalist mobile robotics, vol. 5. Elsevier (2012)
4. Disser, Y.: Mapping Polygons. PhD dissertation, ETH Zurich (2011), No. 20043
5. Donald, B.R.: On information invariants in robotics. Artificial Intelligence **72**(1-2), 217–304 (1995). https://doi.org/10.1016/0004-3702(94)00024-U
6. Erdmann, M.: Understanding Action and Sensing by Designing Action-Based Sensors. IJRR **14**(5), 483–509 (1995). https://doi.org/10.1177/027836499501400506
7. Ganguli, A., Cortes, J., Bullo, F.: Multirobot rendezvous with visibility sensors in nonconvex environments. IEEE TRO **25**(2), 340–352 (2009). https://doi.org/10.1109/TRO.2009.2013493

8. Kamon, I., Rimon, E., Rivlin, E.: TangentBug: A range-sensor-based navigation algorithm. IJRR **17**(9), 934–953 (1998). https://doi.org/10.1177/027836499801700903
9. Latombe, J.C.: Robot Motion Planning. Springer US (1991). https://doi.org/10.1007/978-1-4615-4022-9
10. LaValle, S.M.: Sensor lattices: Structures for comparing information feedback. In: International Workshop on Robot Motion and Control (RoMoCo). pp. 239–246 (2019). https://doi.org/10.1109/RoMoCo.2019.8787364
11. Li, Y., Song, Z., Zheng, C., Bi, Z., Chen, K., Wang, Y., Ma, J.: FRTree Planner: Robot Navigation in Cluttered and Unknown Environments With Tree of Free Regions. IEEE RA-L **10**(4), 3811–3818 (2025). https://doi.org/10.1109/LRA.2025.3544519
12. Lopez-Padilla, R., Murrieta-Cid, R., Lavalle, S.M.: Optimal gap navigation for a disc robot. In: WAFR. vol. 86, pp. 123–138. Springer (2013). https://doi.org/10.1007/978-3-642-36279-8_8
13. Mujahed, M., Mertsching, B.: The Admissible Gap (AG) Method for Reactive Collision Avoidance. In: IEEE ICRA. vol. 2017-July, pp. 1916–1921. Institute of Electrical and Electronics Engineers Inc. (2017). https://doi.org/10.1109/ICRA.2017.8071093
14. Murphy, L., Newman, P.: Using incomplete online metric maps for topological exploration with the gap navigation tree. In: IEEE ICRA. pp. 2792–2797 (2008). https://doi.org/10.1109/ROBOT.2008.4543633
15. Nasir, R., Elnagar, A.: Gap Navigation Trees for Discovering Unknown Environments. Intelligent Control and Automation **6**(4), 229–240 (2015). https://doi.org/10.4236/ICA.2015.64022
16. Nilles, A.Q., Ren, Y., Becerra, I., LaValle, S.M.: A visibility-based approach to computing non-deterministic bouncing strategies. IJRR **40**(10-11), 1196–1211 (2021). https://doi.org/10.1177/0278364921992788
17. O'Kane, J.M., Lavalle, S.M.: On Comparing the Power of Robots. IJRR **27**(1), 5–23 (2008)
18. Sachs, S., LaValle, S.M., Rajko, S.: Visibility-based pursuit-evasion in an unknown planar environment. IJRR **23**(1), 3–26 (2004). https://doi.org/10.1177/0278364904039610
19. Siméon, T., Laumond, J.P., Nissoux, C.: Visibility-based probabilistic roadmaps for motion planning. Advanced Robotics **14**(6), 477–493 (2000). https://doi.org/10.1163/156855300741960
20. Stiffler, N.M., O'Kane, J.M.: A sampling-based algorithm for multi-robot visibility-based pursuit-evasion. In: IEEE/RSJ IROS. pp. 1782–1789 (2014). https://doi.org/10.1109/IROS.2014.6942796
21. Suri, S., Vicari, E., Widmayer, P.: Simple robots with minimal sensing: From local visibility to global geometry. The International Journal of Robotics Research **27**(9), 1055–1067 (2008)
22. Tovar, B., Murrieta-Cid, R., LaValle, S.M.: Distance-optimal navigation in an unknown environment without sensing distances. IEEE TRO **23**, 506–518 (2007). https://doi.org/10.1109/TRO.2007.898962
23. Xu, R., Feng, S., Vela, P.: Potential Gap: A Gap-Informed Reactive Policy for Safe Hierarchical Navigation. IEEE RA-L **6**(4), 8325–8332 (2021). https://doi.org/10.1109/LRA.2021.3104623
24. Zhang, Y., Shell, D.: An abstract theory of sensor eventification. In: RSS. Delft, Netherlands (2024). https://doi.org/10.15607/RSS.2024.XX.065
25. Zhang, Y., Shell, D.A.: Lattices of sensors reconsidered when less information is preferred. In *IEEE ICRA Workshop on Compositional Robotics: Mathematics and Tools* (2021)