Alignment Sets for Sensor Fusion Against Temporal Misalignment

Daniel Kuhse
TU Dortmund, Germany

Mario Günzel
TU Dortmund, Germany

Harun Teper
TU Dortmund, Germany

Lars Willemsen
TU Dortmund, Germany

Georg von der Brüggen
TU Dortmund, Germany

Jian-Jia Chen
RWTH Aachen University, Germany
TU Dortmund, Germany

Abstract

Sensor fusion algorithms combine data from multiple sensors to produce more accurate and reliable results. However, temporal misalignment between sensors, caused by factors such as clock drift, jitter or networking delays, can significantly degrade fusion quality. Prior work on modeling temporal misalignment in sensor fusion algorithms assumes that in the ideal case all samples should be aligned with the same reference time point. We show that this assumption limits its applicability when samples are intentionally taken at different time points, e.g., when a single sensor is sampled multiple times or when sensors operate at different frequencies.

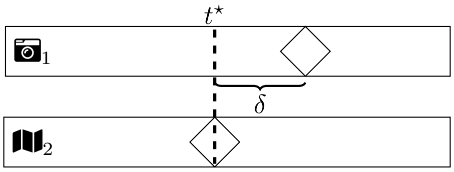
In this paper, we introduce *alignment sets*, which allow system designers to explicitly specify the intended alignment between samples. This flexibility enables more precise temporal misalignment measures that better reflect the actual requirements of sensor fusion scenarios. We prove that alignment sets generalize the prior definitions of temporal misalignment of sensor fusion algorithms. We also provide an evaluation on a camera-LiDAR fusion pipeline for 3D object detection, showing that alignment sets provide more accurate misalignment measures and robustness estimates.

2012 ACM Subject Classification Computer systems organization → Real-time system specification; Computer systems organization → Embedded software

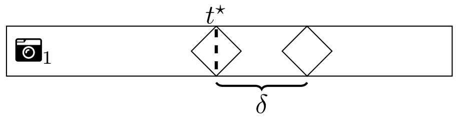
Keywords and phrases Sensor Fusion, Temporal Misalignment, Robustness, Timing Analysis

Digital Object Identifier 10.4230/LIPIcs.ECRTS.2026.8

Funding This work has received funding by the German Federal Ministry of Education and Research (BMBF) in the course of the 6GEM research hub under grant number 16KISK038. This result is part of a project (PropRT) that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 865170). This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project No. 569077889 (PEACH). This work has been funded by the Federal Ministry of Research, Technology and Space (BMFTR) via the *6GEM+* transfer hub under funding references 16KIS2412.



(a) Reference-point-based temporal misalignment. The reference point $t^\star$ is the ideal sampling time for both sensors. Sensor S_1 samples at time $t^\star + \delta$, while sensor S_2 samples at the reference point $t^\star$. The temporal misalignment of sensor S_1 is δ, while the temporal misalignment of sensor S_2 is 0.



(b) Reference-point-based temporal misalignment fusing two samples of the same sensor. Even if both samples are correctly synchronized, since they are not captured at the same time point, there is misalignment, independent from the choice of reference point.

Figure 1 Reference-point-based temporal misalignment for different sensors (left) and the same sensor (right).

1 Introduction

Many cyber-physical systems, like autonomous vehicles and robotic systems, observe the environment with sensors and act based on the perceived data. In many situations, data from multiple sensors, which are designed for different purposes, must be combined to achieve a complete picture of the environment; a process referred to as *sensor fusion* [4]. For example, the 3D location of objects is often estimated by combining a LiDAR sensor's depth information with semantic information derived from camera images.

It is essential for the system's safety that the reliability and performance of sensor fusion can be guaranteed, since incorrect fusion may result in wrong decisions with severe consequences. The *robustness* metric refers to the resilience of a fusion function to small input perturbations. It allows estimating the correctness and reliability of a process. Robustness towards noise [3, 15] or adversarial attacks [21, 27, 29] has been studied.

Temporal misalignment occurs when the sensor data used for fusion are not based on the same physical time (e.g., due to clock drift, jitter, network delays in distributed systems, or varying processing times). It is generally acknowledged as a problem that highly impacts fusion quality and therefore needs to be mitigated [1]. Methods for temporal synchronization are comparatively well studied [5, 17, 20, 22, 31]. De Silva et al. [5] look into spatio-temporal calibration, though they mainly focus on spatial calibration. They experimentally note that bad temporal calibration can lead to substantial errors, but do not aim to give guarantees. These methods are complementary to work on providing robustness guarantees.

While temporal misalignment is a well-known practical issue [1], the first formal definitions and methods to analyze the robustness of sensor fusion algorithms against it have only recently been introduced by Kuhse et al. [16]. Their definitions model misalignment by establishing a single *reference point*, defining a sensor's misalignment as the distance of its actual sample time to this reference point. Finkenzeller et al. [11] build on top of their definition to analyze the robustness of sensor fusion against desynchronization attacks. Sinnema and Maggio [26] further study the impact of temporal misalignment on control systems in a formal manner. With temporal misalignment potentially being a major vulnerability of sensor fusion, they are also a potential attack vector for adversarial attacks on sensor fusion [11, 25].

Figure 1a illustrates the key concept of the reference-point model by Kuhse et al. [16]. It explicitly assumes that, in the ideal case, *all* sensor samples are taken at the *same* reference point $t^\star$. Yet, in some scenarios samples are *intentionally* taken at different time points by design. For example, samples are not expected to be taken at the same time point (i) when fusing two samples from the same sensor, or (ii) when sensors operate at different frequencies. Hence, there is an intended offset between samples and assuming the same reference point for all samples leads to an overestimation of the actual temporal misalignment, which in turn produces overly pessimistic robustness guarantees, as shown in Figure 1b.

A related area is Signal Temporal Logic (STL) [18], which allows specifying the behavior of signals, such as requiring a predicate to hold within some time after another predicate becomes true. A central topic in STL is its robustness, which quantifies the distance of a signal to the boundary between the set of signals satisfying a formula and the set violating it [9]. Several works and tools, such as Breach, provide practical means of computing this robustness *approximately*, with two formulas equivalent under Boolean semantics potentially having different robustness values [6–8,10]. Rino et al. [23,24] recently showed that computing the *precise* temporal robustness of a signal with respect to a formula is NP-hard in general, though a useful fragment (i.e., a restricted subset of the logic) exists for which the complexity is linear regarding signal and formula size. While complex alignment requirements could be expressed in STL, this motivates a specialized framework for sensor fusion robustness.

Our Contributions. We observe that not all sensor samples have the same alignment requirements. What constitutes the *right* alignment can be highly dependent on the specific application. Therefore, a more flexible and precise way of defining temporal misalignment is needed to express these alignment requirements. Towards this, we introduce **alignment sets**, a method that allows system designers to explicitly define the alignment constraints between samples. This enables more accurate definitions of temporal misalignment and robustness for sensor fusion. *Alignment sets* can account for intended offsets, different alignment requirements between different samples, and an invariance to shared time shifts between samples. Specifically, we provide the following contributions:

- In Section 4, we introduce *alignment sets* and use them to define temporal misalignment measures and temporal robustness notions that capture the intended timing requirements of a sensor fusion setup more precisely.
- In Section 5, we show that the resulting alignment-set-based robustness notion generalizes the temporal robustness definitions by Kuhse et al. [16].
- In Section 6, we evaluate alignment sets on a camera-LiDAR fusion setup for 3D object detection and show that they yield more precise temporal misalignment measures and robustness guarantees than the definitions by Kuhse et al. [16].

2 Motivation: Temporal Misalignment in Camera-LiDAR Fusion

Prior work on modeling temporal misalignment in sensor fusion algorithms is based on the notion of a *reference point* [16]. The core assumption is that, in the ideal case, all sensors sample at the same reference time point $t^\star$. Misalignment is then defined as the maximum amount of time by which any sensor's sample deviates from this reference point. This applies to both the *reference-point-based* (see Section 5.1 for details) and *sample-point-based* temporal robustness definitions (see Section 5.2 for details) by Kuhse et al. [16], which only differ in how the reference point is chosen: either as a fixed point in time (reference-point-based) or as a point within a range of the samples (sample-point-based).

To demonstrate the shortcomings of these temporal misalignment definitions and the need for more accurate definitions to quantify temporal misalignments, we consider a camera-LiDAR 3D-object-detection pipeline. The fusion mechanism combines the 2D bounding boxes based on images from the camera with the 3D bounding boxes from the LiDAR point cloud to produce final 3D bounding boxes.

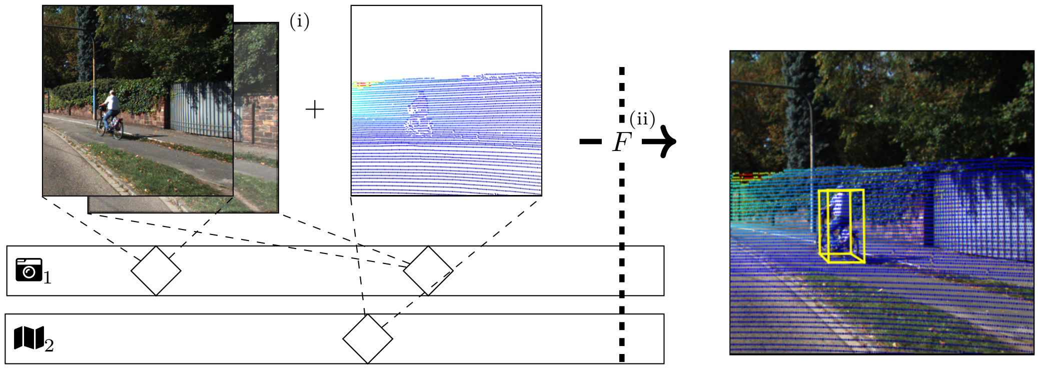


Figure 2 A fusion of two camera samples with a single LiDAR sample.

In practice, camera and LiDAR sensors often operate at different frequencies. A common scenario is that the camera offers a higher frame rate than the LiDAR. Suppose that camera sensors operate at 10 Hz and the LiDAR sensor operates at 5Hz. Hence, if no data should be dropped, multiple camera images must be fused with a single LiDAR point cloud. One possibility is to adopt an interpolation approach, where (i) the bounding-box data of *two* camera images are matched and averaged, and then (ii) are fused with a *single* LiDAR point cloud. This configuration of sensor fusion is depicted in Figure 2.

Hence, the assumption that all sensors should sample at the exact same reference time point does not hold for the camera- LiDAR 3D-object-detection pipeline, since there is always an offset between the two camera samples. One way to still apply the temporal misalignment in [16] is to choose the same reference point for the two camera samples. However, this can lead to very pessimistic estimation of the actual temporal misalignment, as shown below.

▶ **Example 1.** *As the camera samples at* $10Hz$*, two camera images are* $100ms$ *apart. The reference point* $t^\star$ *that minimizes the maximum temporal distance to both images is* $50ms$ *away from each image, as shown in Figure 3. This results in an imprecise measure of temporal misalignment even if the system works perfectly, as it is designed to have two sampling points.*

Let us consider that the camera does not sample perfectly with a 10Hz frequency. Specifically, let us assume that each sampling can drift away from the sampling point by ±1ms, as shown in Figure 3. In this case, the reference point that minimizes the temporal distance to both images is 51ms away from these two images. Using a reference point cannot differentiate the temporal misalignment caused by the temporal drift (jitter) of the sampling points and the offset due to the interpolation of two images from different sampling points.

Therefore, the approach by Kuhse et al. [16] cannot capture the example without overestimation, nor can it differentiate definition-induced misalignment from actual misalignment. Hence, we need definitions and models that can express requirements on how samples should be aligned. We develop this step-by-step, starting with an extended sensor fusion model that supports multiple samples per sensor is introduced in Section 3 and culminating in our proposed definitions based on *alignment sets* in Section 4.

3 Sensor-Fusion Model

We consider a system with n sensors $S_1, \ldots, S_n$. Each sensor produces a stream of measured values at given time points, and we denote the sensor data of sensor S_i at given time point t as $S_i(t)$. Formally, S_i is defined as a function from the time domain $\mathbb{R}_{\geq 0}$ to the space of measurement values $\mathbb{X}_i$. Specifically, $S_i : \mathbb{R}_{\geq 0} \to \mathbb{X}_i$.

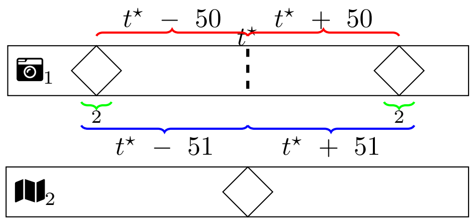


Figure 3 For a reference time point $t^\star$ in the middle between the two camera samples, reference-point based temporal misalignment definitions quantifies the misalignment of a perfectly working system with 50ms (red). When allowing a jitter of $\pm$1ms (green) to each camera sample, the misalignment is given as 51ms (blue).

Considering our motivational example in Section 2, there is one camera sensor S_{cam} and one LiDAR sensor S_{LiDAR}. After initial processing, the camera sensor produces 2D bounding boxes as measurements, while the LiDAR sensor produces 3D bounding boxes. Thus, we have $\mathbb{X}_{\text{cam}}$ as the space of 2D bounding boxes and $\mathbb{X}_{\text{LiDAR}}$ as the space of 3D bounding boxes. The k-th sample of sensor S_i is taken at a specific time point $t_{i,k} \in \mathbb{R}_{\geq 0}$, denoted as $S_i(t_{i,k})$. The fusion function takes a fixed number m_i of samples from each sensor S_i as input.

We collect all timestamps used in a specific fusion into a *timestamp tuple*:

$$T = (T_1, \ldots, T_n) \in \mathbb{T} \quad \text{with} \quad \mathbb{T} = \mathbb{R}_{\geq 0}^{m_1} \times \cdots \times \mathbb{R}_{\geq 0}^{m_n}$$

where $T_i = (t_{i,1}, \ldots, t_{i,m_i}) \in \mathbb{R}_{\geq 0}^{m_i}$ is the timestamp vector for S_i and $\mathbb{T}$ is the space of all sample timestamp tuples that could be fused (without assumptions about timestamp order).

For our motivational example, we have $m_{\text{cam}} = 2$ and $m_{\text{LiDAR}} = 1$ (i.e., two camera samples and one LiDAR sample are fused). The timestamp tuple is $T = (t_{\text{cam},1}, t_{\text{cam},2}, t_{\text{LiDAR}})$.

Let $\mathcal{S} = \{(S_i, k) \mid i \in \{1, \ldots, n\},\, k \in \{1, \ldots, m_i\}\}$ denote the set of all sensor samples involved in a fusion. For a sample $s \in \mathcal{S}$, we write t_s for its timestamp. The concrete sensor samples in the motivational example are $\mathcal{S} = \{(S_{\text{cam}}, 1), (S_{\text{cam}}, 2), (S_{\text{LiDAR}}, 1)\}$.

Underlying the fusion is a *sensor-fusion algorithm* F that takes the collected sensor samples and produces a fused output in a space $\mathbb{Y}$:

$$F : \mathbb{X}_1^{m_1} \times \cdots \times \mathbb{X}_n^{m_n} \to \mathbb{Y}$$

For our study of temporal misalignment, we consider F as an arbitrary function (black box).

For our motivational example, the fusion algorithm F is composed of an averaging operation F_{avg} combining the two camera detections cam_1, cam_2 and a projection-based fusion function $F_{\text{projection}}$ that combines this average with the LiDAR detection $LiDAR$: $F(cam_1, cam_2, LiDAR) = F_{\text{projection}}\big(F_{\text{avg}}(cam_1, cam_2),\, LiDAR\big)$.

To describe the temporal misalignment and its impact on the robustness of the sensor fusion algorithm, we only need to consider how the fusion output changes relative to changes in the timestamps of the sensor samples. We therefore define a *fusion function* $f : \mathbb{T} \to \mathbb{Y}$ that maps the timestamps of the sensor samples to the fused output. Given a function $sample : \mathbb{T} \to \mathbb{X}_1^{m_1} \times \cdots \times \mathbb{X}_n^{m_n}$ that maps the timestamps to the corresponding sensor samples, the fusion function is defined as:

$$f(T) = F(sample(T))$$

For example, in our camera-LiDAR fusion scenario, the fusion function f is defined as:

$$f((t_{\text{cam},1}, t_{\text{cam},2}, t_{\text{LiDAR}})) = F(sample((t_{\text{cam},1}, t_{\text{cam},2}, t_{\text{LiDAR}})))$$

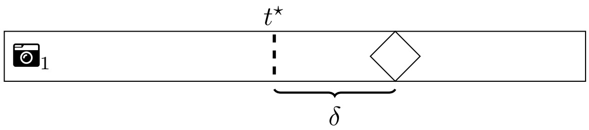


Figure 4 Anchored alignment: a sample from a sensor s should align to a fixed time point $t^\star$ (called anchor). The actual sample occurs at $t^\star + \delta$, inducing a misalignment of δ.

All subsequent definitions of temporal misalignment and robustness against temporal misalignment will use this fusion function f, avoiding the need to explicitly refer to the underlying fusion algorithm F or the sensor samples.

4 Misalignment Functions

In this section, we provide a new definition of temporal misalignment based on *alignment sets*. As discussed in Section 2, prior definitions [16] of temporal misalignment assume that perfect alignment means that all samples are taken at the same moment in time, an assumption that is too restrictive. *Alignment sets* are a more flexible mechanism of quantifying misalignment, which allows more precise measurement of misalignment and robustness against it.

The core idea of *alignment sets* is that the meaning of *aligned* depends on the specific application and fusion mechanism. *Alignment sets* are a structured way to specify these alignment requirements. In Section 4.1, we introduce *alignment constraints*, the building blocks of alignment sets, which specify what perfect alignment means for individual sensor samples. Multiple *alignment constraints* form an *alignment set* (Section 4.2), which specifies the intended synchronization for a fusion function. Individual constraints have a corresponding measure of how far an actual timestamp deviates from the ideal case. Aggregating these individual constraint deviations obtains an overall misalignment for a given timestamp tuple T. We provide a definition of temporal robustness based on alignment sets in Section 4.3.

4.1 Alignment Constraints

Alignment constraints specify the intended synchronization at the level of individual sensor samples. *Misalignment* is how far the actual timestamp of a sample deviates from the intended synchronization specified by the alignment constraint. An individual alignment constraint describes where the time point t_s of a sample s should ideally be, by specifying the relationship of t_s to other sample timestamps or fixed time points. Given a timestamp tuple T, one can then quantify how far t_s deviates from this ideal case as a residual.

We first provide examples of common alignment scenarios, based on the motivational example from Section 2, and observe that earlier examples are special cases of later examples. We provide a unifying formal definition, showing how it captures all these scenarios.

Scenario 1: Alignment to a fixed time point. The *actual* time point of a sample s should be aligned with a fixed, *ideal* time point $t^\star$. We call such a fixed time point an **anchor**. This ideal time point purely specifies when (with perfect alignment) s should be sampled. We denote such a constraint as $s \sim t^\star$. Given the actual timestamp t_s, this constraint induces a resulting misalignment $t_s - t^\star$. Figure 4 illustrates anchored alignment.

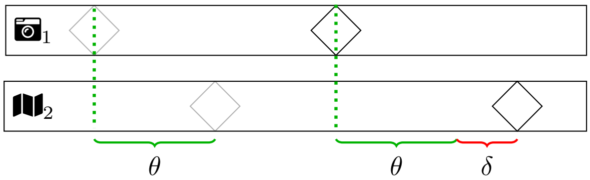


Figure 5 Alignment to another sample for two sensors S_1 and S_2 with samples aligned with an offset θ. Only when disproportionate shifts occur is the misalignment δ affected.

Such a constraint is similar to the notion of the temporal misalignment defined by Kuhse et al. [16] (illustrated in Figure 1a), but with the important difference that the reference point $t^\star$ is specified separately for each constraint. For instance, this scenario allows to specify that two camera samples should be taken at different time points.

Scenario 2: Alignment to another sample. Here, the *relative* alignment between samples is more important. That is, the exact time points of the samples may not be important, as long as they are aligned with each other in the right way. Hence, if both samples are shifted by the same amount, the fusion algorithm still works perfectly. We denote such an alignment as $s_2 \sim s_1 + \theta$, specifying that $s_2 \in \mathcal{S}$ should occur at offset $\theta \in \mathbb{R}$ from a reference sample $s_1 \in \mathcal{S}$, with a resulting misalignment $t_{s_2} - t_{s_1} - \theta$. Figure 5 illustrates this scenario for two sensors S_1 and S_2 with samples s_1 and s_2.

For instance, in our motivational example, the second camera sample should be taken at a specific offset θ from the first camera sample, dependent on the camera frame rate.

Scenario 3: Alignment to a combination of samples. The alignment constraint does not depend on a single sample but on the combination of multiple samples. For instance, in our motivational example, the LiDAR sample is fused with the average of the two camera samples. Hence, the fusion algorithm also assumes that the LiDAR sample is aligned in the middle of the two camera samples. This is equivalent to aligning to a weighted combination of both camera samples, which we denote as $s \sim \frac{1}{2}s_1 + \frac{1}{2}s_2$, with a misalignment $t_s - \frac{1}{2}t_{s_1} - \frac{1}{2}t_{s_2}$.

Based on this, we provide a unifying formal definition of alignment constraints, which captures all the above scenarios as special cases.

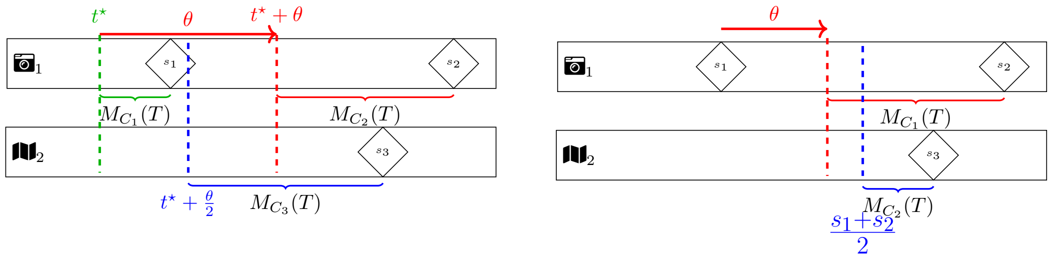
▶ **Definition 2** (Alignment Constraint). *An **alignment constraint** C specifies that a sample $s \in \mathcal{S}$, the **constrained sample**, should occur at a weighted combination of **reference samples** plus an offset:*

$$C:\ s \sim \left(\sum_{r \in R} w_r \cdot r\right) + \theta \tag{1}$$

*where $R \subseteq \mathcal{S}$ is a set of reference samples, $w_r \in \mathbb{R}$ are weights, and $\theta \in \mathbb{R}$ is an offset. The **induced misalignment** $M_C(T)$ of constraint C given a timestamp tuple T is:*

$$M_C(T) = t_s - \left(\sum_{r \in R} w_r \cdot t_r\right) - \theta \tag{2}$$

Alignment to an anchor corresponds to $R = \emptyset$ with $\theta = t_s^\star$. Alignment to another sample corresponds to $R = \{r\}$ with $w_r = 1$. Alignment to a midpoint between r_1 and r_2 corresponds to $R = \{r_1, r_2\}$ with $w_{r_1} = w_{r_2} = \frac{1}{2}$ and $\theta = 0$.



(a) Anchored alignment. The alignment set $\mathcal{A}$ for three samples s_1, s_2, s_3 aligned to three anchors $t^\star$ (dotted green), $t^\star + \theta$ (dotted red), $t^\star + \frac{\theta}{2}$ (dotted blue), respectively. The alignment constraints are $C_1 : s_1 \sim t^\star$ (green), $C_2 : s_2 \sim t^\star + \theta$ (red), and $C_3 : s_3 \sim t^\star + \frac{\theta}{2}$ (blue), alongside their misalignment $M_{C_1}(T)$, $M_{C_2}(T)$, and $M_{C_3}(T)$.

(b) Relative alignment. An alignment set $\mathcal{A}$ for three samples s_1, s_2, s_3 where no anchors are used, with constraints $C_1 : s_2 \sim s_1 + \theta$ (red), and $C_2 : s_3 \sim \frac{1}{2}s_1 + \frac{1}{2}s_2$ (blue), alongside their induced misalignment $M_{C_1}(T)$ and $M_{C_2}(T)$ for an example timestamp tuple T.

Figure 6 Anchored vs. Relative alignment sets.

4.2 Alignment Sets

A single alignment constraint specifies the intended timing of one sensor sample. However, for more complex fusion functions, a single alignment constraint is often insufficient to capture the intended synchronization. Multiple constraints together are needed to describe the intended synchronization. In the example from Section 2, both the relationship (i) between the two camera samples, and (ii) between the camera samples and the LiDAR sample are relevant for synchronization. *Alignment sets* combine multiple alignment constraints into a specification of how all sensor samples involved in a fusion function should ideally be aligned.

Formally, we define alignment sets as finite sets of alignment constraints over samples $\mathcal{S}$.

▶ **Definition 3** (Alignment Set). *An* ***alignment set*** *is a finite set $\mathcal{A}$ of alignment constraints on the sample set $\mathcal{S}$.*

We call an alignment set ordered if it has no cyclic dependencies between samples.

▶ **Definition 4** (Ordered Alignment Set). *An alignment set $\mathcal{A}$ is called* ***ordered*** *if it admits an ordering $C_1, \ldots, C_{|\mathcal{A}|}$ such that for any constraint $C_i : s_i \sim \left(\sum_{r \in R_i} w_r \cdot r\right) + \theta_i$, each reference sample $r \in R_i$:*

- *is* ***unconstrained*** *(i.e., r is not the constrained sample of any constraint in $\mathcal{A}$), or*
- *is* ***constrained earlier*** *(i.e., r is the constrained sample of a constraint C_j with $j < i$ in the ordering).*

We provide *two* examples of alignment sets for the camera-LiDAR fusion scenario from Section 2. The exact alignment requirements depend on the specific application. Depending on the application's requirements, it is therefore possible that multiple different alignment sets are reasonable for the same fusion function. A key question is whether the perfect alignment is *anchored* to specific time points or whether only their relative timing matters.

▶ **Example 5** (Anchored Alignment Set). If the fusion expects the data to reflect a specific point in time, then perfect alignment requires that the samples are taken at specific, predetermined time points. Suppose that the temporal alignment requires that the first camera sample should be taken at time $t^\star$, the second camera sample at time $t^\star + \theta$ (where θ is determined by the camera frame rate), and the LiDAR sample should be taken at the midpoint between the two anchors, which is $t^\star + \frac{\theta}{2}$. This is captured by the alignment set

$$\mathcal{A}_{\text{anchored}} = \left\{ C_1 \colon s_1 \sim t^\star, \quad C_2 \colon s_2 \sim t^\star + \theta, \quad C_3 \colon s_3 \sim t^\star + \frac{\theta}{2} \right\}$$

which is illustrated in Figure 6a, showing the constraints and their induced misalignments for an example timestamp tuple T. All three samples are aligned to anchors. Hence, low misalignment means that the fusion result corresponds to these specific time points.

▶ **Example 6** (Relative Alignment Set). Assume it is sufficient that the samples are aligned with each other, but not to specific time points. For instance, assume the two camera samples should be aligned with an offset θ and the LiDAR sample should be aligned in the middle of the two camera samples. The resulting alignment set is:

$$\mathcal{A}_{\text{relative}} = \{C_1 \colon s_2 \sim s_1 + \theta, \quad C_2 \colon s_3 \sim \tfrac{1}{2}\, s_1 + \tfrac{1}{2}\, s_2\},$$

which is illustrated in Figure 6b, showing the constraints and their induced misalignments for an example timestamp tuple T.

4.3 Robustness Against Temporal Misalignment

For an alignment set $\mathcal{A}$, we now explain how to evaluate the misalignment function and the robustness against temporal misalignment constrained by the alignment set. We start from a general definition of a misalignment function.

▶ **Definition 7** (Misalignment Function). *Given an alignment set $\mathcal{A}$ and an aggregation function $g : \mathbb{R}^{|\mathcal{A}|} \to \mathbb{R}_{\geq 0}$, $\mathcal{A}$'s **misalignment function** $M_{\mathcal{A}} : \mathbb{T} \to \mathbb{R}_{\geq 0}$ is defined as:*

$$M_{\mathcal{A}}(T) = g\big((M_C(T))_{C\in\mathcal{A}}\big) \tag{3}$$

where $(M_C(T))_{C\in\mathcal{A}}$ is a vector of individual misalignments for all constraints in $\mathcal{A}$.

The function g can be any statistical functions or extremum functions. We give a bound on the time complexity of evaluating this misalignment measure $M_{\mathcal{A}}(T)$.

▶ **Lemma 8.** *Given an alignment set $\mathcal{A}$ and timestamp tuple T, the misalignment function $M_{\mathcal{A}}(T)$ operates in $O(G(|\mathcal{A}|) + |\mathcal{A}| \cdot k)$ time complexity, where $|\mathcal{A}|$ is the cardinality of the alignment set $\mathcal{A}$, k is $\max_{C\in\mathcal{A}} |R_C|$, and $O(G(|\mathcal{A}|))$ is the time complexity to evaluate the function $g\big((M_C(T))_{C\in\mathcal{A}}\big)$.*

Proof. The misalignment $M_C = t_s - \sum_{r\in R_C}(w_r \cdot t_r) - \theta_C$ of a single constraint is a weighted sum over at most k referenced samples, with a time complexity of $O(k)$. The complete misalignment vector $(M_C(T))_{C\in\mathcal{A}}$ has therefore a time complexity of $O(|\mathcal{A}| \cdot k)$. Aggregating this through g then adds $O(G(|\mathcal{A}|))$, giving a total time complexity of $O(G(|\mathcal{A}|) + |\mathcal{A}| \cdot k)$. ◀

In case the aggregation function g operates in linear time, which applies to simple aggregation functions like extremum functions and combinations of them, the time complexity of Lemma 8 is then $(|\mathcal{A}| \cdot k)$, used in our experiments in Section 6. A finer analysis accounting for the geometry of constraints from g is left to future work (see Section 7).

A fusion function is robust against temporal misalignment when a small temporal misalignment only cause a small error in the fusion output. Given a misalignment function $M_{\mathcal{A}}$, we formalize this intuition as follows: whenever the misalignment measured by $M_{\mathcal{A}}(T)$ is at most Δ, the fusion error quantified by some error function E should be at most ϵ.

▶ **Definition 9** (ϵ-Δ-Temporal Robustness). *Let $M_{\mathcal{A}} : \mathbb{T} \to \mathbb{R}_{\geq 0}$ be a misalignment function for an alignment set $\mathcal{A}$ that quantifies the temporal misalignment between sensor timestamps. Let $E : \mathbb{T} \times \mathbb{Y} \to \mathbb{R}_{\geq 0}$ be a function that measures the error of the sensor fusion output.*

For a sensor fusion function $f : \mathbb{T} \mapsto \mathbb{Y}$, error threshold $\epsilon > 0$, and maximum misalignment $\Delta > 0$, we say that f is ϵ-Δ-temporally robust if and only if:

$$\forall T \in \mathbb{T} : (M_{\mathcal{A}}(T) \leq \Delta \implies E(T, f(T)) \leq \epsilon) \tag{4}$$

What constitutes an appropriate error function E heavily depends on the specific application domain and fusion function. As an example, for a camera-LiDAR fusion function, the error function E could measure the deviation of detected object positions from their true position. One robustness guarantee of $\Delta = 10ms$ and $\epsilon = 0.5m$ would then mean that as long as all samples are within 10 ms of their ideal timestamps ($M_{\mathcal{A}}(T) \leq 10ms$), the position error of detected objects is at most 0.5 meters ($E(T, f(T)) \leq 0.5m$).

The robustness guarantee requires a domain specific error function. A natural choice is to compare the fusion output under misaligned timestamp tuples to the output under a timestamp tuple with no misalignment, measuring how much misalignment changes the result. When there are multiple timestamp tuple with no misalignment, some choice needs to be made as to which is compared to.

5 Generalization of Prior Robustness Definitions

In this section we compare our alignment set method to prior definitions of temporal misalignment and robustness, and discuss challenges and limitations. Alignment sets build upon and generalize the previous definitions of temporal misalignment and robustness. We show how the previous definitions [16] can be modeled within the alignment set method, proving that alignment sets are a strict generalization of these previous definitions.

5.1 Reference-point based Temporal Robustness

Reference-point-based temporal robustness as given by Kuhse et al. [16] is based on a single reference time point $t^\star$ and considers misalignments of the sensor samples with respect to this reference point. That is, given a reference time point $t^\star$, analogous to the ϵ-Δ-temporal robustness in Definition 9, the sensor fusion function f is considered robust against temporal misalignment for a misalignment threshold Δ and error threshold ϵ if and only if:

$$\forall t_1, \ldots, t_n \in [t^\star - \Delta, t^\star + \Delta] : L(f(t^\star, \ldots, t^\star), f(t_1, \ldots, t_n)) \leq \epsilon \tag{5}$$

where L is a loss function that measures the difference between the fused output at time t and the fused output at the reference time point $t^\star$.

We note that this loss function L is different from our error function E in Definition 9: L directly compares two fusion outputs, while E quantifies the error of a fusion result given timestamps T. Our E is more flexible, as shown in the proof of Lemma 10, their loss function can be embedded into our error function by defining $E(T, f(T)) = L(f(t^\star, \ldots, t^\star), f(T))$. We chose to generalize the error function as reference-point-based robustness and sample-based robustness (Section 5.2) used L differently, comparing against a fusion at a single reference point and against multiple fusions in a range of reference points, respectively. The error function E allows both of these situations to be captured as special cases, while also allowing for more general error functions that do not necessarily compare against a reference point fusion output (e.g., an error function that directly quantifies the error of the fusion output without comparing it to a reference point fusion output).

We show that alignment set based robustness is a strict generalization via proof by construction, showing that this definition can be captured by an alignment set with appropriate constraints, misalignment measure, and error function.

▶ **Lemma 10.** *There exists an alignment set $\mathcal{A}$, a misalignment measure M, and an error function E such that the alignment set robustness condition:*

$$(\forall T, [M_{\mathcal{A}}(T) \leq \Delta] \Rightarrow E(T, f(T)) \leq \epsilon) \tag{6}$$

is equivalent to the reference-point based robustness definition shown in Eq. 5.

The proof is straightforward and therefore given in Appendix A.

5.2 Sample-based Temporal Robustness

Sample-based temporal robustness as given by Kuhse et al. [16] assumes time stamps for each sensor sample as given, evaluating robustness by considering possible reference points inside the range spanned by the sensor samples.

Formally, given $t_1, \ldots, t_n$ as the timestamps of the sensor samples, two different notions of sample-based temporal robustness are defined, differing in strictness. *Strong* sample-based temporal robustness requires that the fusion output is close to the output at *every* reference point $t^\star$ in the range spanned by the sensor timestamps, while *weak* sample-based temporal robustness only requires that the fusion output is close to the output at *some* reference point $t^\star$ in this range. Strong sample-based temporal robustness means that the fusion output is stable across the entire range of possible reference points to represent all of them, while weak definition only requires that it reflects some point in the range. Formally, strong sample-based temporal robustness is defined as:

$$\forall t^\star \in [\min(t_i), \max(t_i)] : L(f(t^\star, \ldots, t^\star), f(t_1, \ldots, t_n)) \leq \epsilon \tag{7}$$

while *weak* sample-based temporal robustness is defined as:

$$\exists t^\star \in [\min(t_i), \max(t_i)] : L(f(t^\star, \ldots, t^\star), f(t_1, \ldots, t_n)) \leq \epsilon \tag{8}$$

In both cases, L is the same kind of loss function as in Section 5.1.

In the global notion of sample-based temporal robustness, these definitions must hold for all possible timestamp combinations $T = (t_1, \ldots, t_n)$ where the misalignment between the timestamps is bounded by a threshold Δ.

Following the same approach as for the reference-point-based definition, we show that alignment set based robustness is a strict generalization of sample-based robustness by showing that both strong and weak sample-based robustness can be captured by an alignment set with appropriate constraints, misalignment measure, and error function.

▶ **Lemma 11.** *There exists an alignment set $\mathcal{A}$, a misalignment measure M, and an error function E for the global variants of respectively the strong and weak sample-based robustness definitions given in Eqns. 7 and 8 such that the alignment set robustness condition:*

$$(\forall T, [M(T) \leq \Delta] \Rightarrow E(T, f(T)) \leq \epsilon) \tag{9}$$

is equivalent to the respective sample-based robustness definition.

The proof of this lemma is given in the Appendix A.

6 Evaluation

We evaluate our alignment set method for modeling temporal misalignment and robustness in the context of camera-LiDAR fusion for 3D object detection. Our focus is showing how alignment sets in several scenarios with different synchronization requirements lead to more precise measures of temporal misalignment than reference-point based definitions from prior work [16]. This in turn leads to more informative robustness estimates. For our evaluation, we use the KITTI dataset [12], which has been used by both Kuhse et al. [16] and Finkenzeller et al. [11] to evaluate temporal robustness for camera-LiDAR fusion where one camera frame is fused with one LiDAR scan. We extend this to the scenario introduced in Section 2 where two camera frames are fused with one LiDAR scan. The full evaluation setup, including tested alignment sets, is described in Section 6.1 and Section 6.2. How robustness is evaluated is described in Section 6.3. Section 6.4 presents the results for three different scenarios for a single fusion of two camera samples and a LiDAR sample, showing the difference in misalignment and robustness estimates between different alignment sets. Section 6.5 considers alignment sets for a full 8-minute drive, showing that alignment sets can be applied to a more complex scenario with many samples and constraints.

6.1 Experimental Setup

Our evaluation focuses on showing how alignment sets can more precisely capture the temporal misalignment of the camera-LiDAR scenario from Section 2. This in turn leads to robustness estimates with tighter error bounds. In this section, we describe the fusion pipeline, evaluation scenarios for a single fusion, dataset, and metrics. The full-drive setup (Section 6.5) reuses the same fusion pipeline but defines its own scenarios.

The camera-LiDAR fusion we consider is based on the fusion algorithm provided in Autoware Universe's perception stack [14]. Camera images are processed by YOLOv9 [28] into 2D bounding boxes, while LiDAR point cloud data is processed by a CenterPoint [32] network provided by Autoware, producing 3D bounding boxes. Camera and LiDAR often operate at different frequencies, e.g., the camera might operate at a higher frequency. To use all available information, multiple camera images are fused with one LiDAR scan. Each camera image is processed independently, the 2D bounding boxes are matched by bounding box overlap, and their coordinates averaged. These 2D bounding boxes are then fused with the 3D bounding boxes by projecting the latter into the camera plane and matching with the 2D bounding boxes by overlap. Only matched 3D bounding boxes are kept, with the class labels taken from the 2D detections. We run the inference of the YOLOv9 and CenterPoint models on an NVIDIA RTX 4090 GPU, with a single fusion taking about 56 ms.

Our full evaluation pipeline, including the sampling of misaligned timestamp tuples, misalignment computation, and aggregation of fusion errors, is implemented in custom Python code, with no external verification or analysis tool used.

We evaluate the temporal misalignment and robustness of this fusion pipeline under two configurations: In the first (**LiDAR centered**) the LiDAR sample is intended to be taken in the middle of the two camera samples, while in the second (**LiDAR late**) the LiDAR sample is intended to be taken at the same time as the second camera sample.

For these configurations, we consider three different scenarios:

- **Synchronized**: Samples are taken exactly at the intended time points according to the configuration.
- **Jitter**: Samples independently deviate from their intended time point by a random amount up to $\pm j$ ms, where j is the maximum jitter.
- **Uniform shift**: All sensor samples are shifted by an identical constant offset of d ms, where d is the shift amount.

In all scenarios, we determine the timestamp misalignment for the reference-point-based definition from [16] and our alignment sets (Section 6.2), showing that the choice of alignment set significantly affects the reported misalignment and resulting robustness estimates.

The evaluation is performed on the KITTI dataset [12], which provides synchronized camera and LiDAR data at 10 Hz (i.e., every 100 ms) across 155 drives. We use the raw dataset, which contains full series of camera frames and LiDAR scans with their corresponding timestamps, allowing us to simulate various temporal misalignment scenarios by shifting timestamps and selecting appropriate samples, as in prior work [11,16]. Since KITTI provides samples every 100 ms, both jitter and offset are simulated by selecting neighboring samples, so the applied jitter or offset is necessarily a multiple of 100 ms. For the full run in Section 6.5, we remove this restriction by interpolating data in steps of 5 ms.

For the **LiDAR centered** configuration, the LiDAR sample should lie at the midpoint between two camera frames. Since KITTI only provides samples for both sensors every 100 ms, no LiDAR sample exists at the midpoint (50ms) between consecutive camera frames. Thus, we pair camera frames that are 200 ms apart (i.e., separated by one frame, e.g., 0ms and 200ms, 100ms and 300ms, ...), fusing each pair with the LiDAR sample between them. For consistency, we keep the sampling of camera images the same for the **LiDAR late** configuration, but choose the LiDAR sample that aligns with the later camera sample.

6.2 Alignment Sets for Evaluation

We examine five ways to specify alignment. In addition to the reference-point-based definition, we consider four different alignment sets, namely one anchored and one relative alignment for both the LiDAR centered and LiDAR late configurations, capturing different synchronization requirements for the camera-LiDAR fusion setup.

Using the notation from Section 3, the sensor samples are $c_1 = (S_{\text{cam}}, 1)$, $c_2 = (S_{\text{cam}}, 2)$, and $l_1 = (S_{\text{lidar}}, 1)$, with θ denoting the time difference between the two camera frames. Recall that $s \sim r + \theta$ means sample s is intended to be taken at time $r + \theta$ (see Definition 2). For anchored alignment sets, we use $t^\star$ as a fixed reference time point serving as the anchor. Note that in the relative alignment sets, c_1 is unconstrained since only the relative timing between samples matters, not their absolute position.

Anchored: LiDAR centered. All samples are anchored, the LiDAR sample is aligned in the middle between the two camera frames: $\mathcal{A}_{\text{anchored-inter}} = \{c_1 \sim t^\star,\ c_2 \sim t^\star{+}\theta,\ l_1 \sim t^\star{+}\frac{\theta}{2}\ \}$

Anchored: LiDAR late. All samples are anchored, the LiDAR sample is aligned with the second camera frame: $\mathcal{A}_{\text{anchored-late}} = \{c_1 \sim t^\star,\ c_2 \sim t^\star + \theta,\ l_1 \sim t^\star + \theta\}$

Relative: LiDAR centered. The second camera frame is relatively aligned to the first one, and the LiDAR sample is temporally centered relative to the two camera frames: $\mathcal{A}_{\text{relative-inter}} = \{c_2 \sim c_1 + \theta,\ l_1 \sim \frac{1}{2}c_1 + \frac{1}{2}c_2\}$

Relative: LiDAR late. The second camera frame is aligned to the first, and the LiDAR sample is aligned with the second camera frame: $\mathcal{A}_{\text{relative-late}} = \{c_2 \sim c_1 + \theta,\ l_1 \sim c_2\}$

For all alignment sets we use the **maximum absolute misalignment** across the constraints as the misalignment measure (Definition 7). That is, $g(m_1, \ldots, m_{|\mathcal{A}|}) = \max_i |m_i|$ where m_i is the misalignment of constraint C_i.

6.3 Robustness Evaluation

We evaluate the robustness estimates of all five specifications: the alignment sets presented in Section 6.2, in comparison to the **reference-point-based** temporal robustness definitions by Kuhse et al. [16].[1] For each misalignment specification and each misalignment threshold Δ (from Definition 9), we evaluate the error of the fusion output using 77500 timestamp tuples (500 for each drive in KITTI), sampled from the entire KITTI dataset. Generating all these samples takes less than a second, with the fusion evaluation being the bottleneck, taking about one hour per alignment set and Δ combination. Evaluating the fusion error of one individual sample takes about 0.05 seconds. Peak memory consumption is 3.3 GB.

[1] We do not evaluate the sample-point-based definitions [16], as their error functions compare against multiple reference points (all points in the range spanned by the timestamps), making them not directly comparable to the other definitions, which all compare against a single timestamp tuple with no misalignment. A fair comparison would require a separate evaluation, where for each alignment set one would also compare against multiple timestamp tuples with no misalignment.

As the KITTI dataset consists of only synchronized camera-LiDAR series, we first describe how we sample misaligned timestamp tuples from the synchronized KITTI data in Section 6.3.1, then explain how we evaluate the resulting fusion error in Section 6.3.2.

6.3.1 Sampling Misaligned Timestamp Tuples

We sample timestamp tuples with temporal misalignment at most Δ as follows:

- For reference-point-based misalignment, we sample a reference point $t^\star$ from the available KITTI timestamps and each sensor timestamp independently from $[t^\star - \Delta, t^\star + \Delta]$.
- For each alignment set $\mathcal{A}$ introduced in Section 6.2, we follow the ordering of the constraints (see Definition 4). First we sample $t^\star$ for the anchored sets. For the relative sets, we sample the timestamp of the unconstrained sample (c_1).

  For each alignment constraint C_i, we sample a misalignment value $|m_i| \leq \Delta$ and compute the sample's timestamp as $t_{s_i} = \sum_{r \in R_i} w_r \cdot t_r + \theta_i + m_i$. This operation is always successful since all times t_r that are referenced in the alignment constraint must have already been determined before due to the ordering. Since each constraint misalignment is at most Δ, it follows that $M_\mathcal{A}(T) \leq \Delta$.

  This sampling procedure has a runtime complexity of $O(|\mathcal{A}| \cdot k)$ per tuple, where k is the maximum number of samples referenced in a constraint (see Lemma 8). Computing each sample's timestamp requires $O(k)$ operations, with $|\mathcal{A}|$ such steps required. Thus, generating timestamp tuples is comparatively inexpensive relative to the fusion evaluation.

6.3.2 Error Evaluation

To determine the error of a fusion output we compare to the output under *perfectly aligned timestamp tuples*, tuples with no misalignment, defined as follows:

- For the reference-point specification, the perfectly aligned timestamp tuple is uniquely determined by the reference point (i.e., all sensor timestamps are set to $t^\star$).
- For anchored alignment sets, the perfectly aligned timestamp tuple is uniquely determined by the anchors.
- For relative alignment sets, we choose the perfectly aligned tuple where t_{c_2} matches the second camera sample in the misaligned timestamp tuple.

We use two different error measures, like in the prior work by Kuhse et al. [16]:

- **Position error**: the Euclidean distance between the detected positions of an object.
- **F1 score**, defined as $F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$, where precision is the percentage of correct detections (i.e., match a detected object from the perfectly aligned fusion), and recall is the percentage of objects from the perfectly aligned fusion that is detected. F1 score combines them into a single metric, penalizing both false positives and false negatives.

For both error measures, we report the average values and the standard deviations. For the position error we also report the maximum. Even for low misalignments the F1 score is occasionally zero, which is why we do not report the worst error for it.

These error measures instantiate the error function E from Definition 9 by comparing the fusion output under misaligned timestamps to the output under perfectly aligned timestamps, measuring how much misalignment degrades the result.

Table 1 Full robustness evaluation, showing for each specification and misalignment threshold Δ the mean, standard deviation, and maximum position error, as well as the mean and standard deviation of the F1 score.

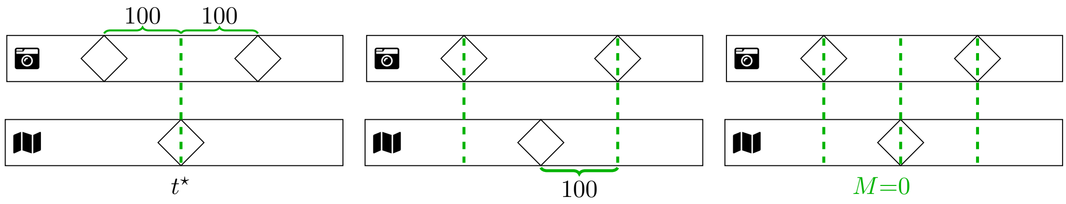
		Position Error (m)			F1 Score	
Specification	**Δ (ms)**	**Mean**	**Std**	**Max**	**Mean**	**Std**
Reference-point	100	0.269	0.35	3.487	0.831	0.22
	200	0.356	0.43	4.497	0.764	0.25
	300	0.383	0.46	4.552	0.722	0.27
	400	0.396	0.48	4.245	0.692	0.28
	500	0.398	0.48	4.630	0.665	0.28
	600	0.399	0.49	5.610	0.646	0.29
$\mathcal{A}_{\text{anchored-inter}}$	100	0.254	0.33	3.484	0.823	0.22
	200	0.340	0.41	3.543	0.782	0.24
	300	0.370	0.45	3.848	0.757	0.26
	400	0.385	0.47	4.164	0.735	0.27
	500	0.388	0.48	4.768	0.715	0.27
	600	0.389	0.48	5.610	0.701	0.28
$\mathcal{A}_{\text{anchored-late}}$	100	0.252	0.33	3.276	0.825	0.22
	200	0.338	0.41	3.543	0.783	0.24
	300	0.370	0.45	4.552	0.754	0.26
	400	0.384	0.46	4.796	0.735	0.26
	500	0.391	0.48	5.495	0.714	0.27
	600	0.388	0.47	5.467	0.701	0.28
$\mathcal{A}_{\text{relative-inter}}$	100	0.307	0.38	4.753	0.808	0.22
	200	0.349	0.42	3.880	0.785	0.24
	300	0.378	0.46	4.545	0.762	0.26
	400	0.397	0.48	6.227	0.743	0.26
	500	0.403	0.49	7.755	0.724	0.27
	600	0.402	0.49	6.155	0.714	0.28
$\mathcal{A}_{\text{relative-late}}$	100	0.321	0.41	3.543	0.801	0.23
	200	0.353	0.45	4.753	0.778	0.25
	300	0.376	0.47	5.304	0.758	0.26
	400	0.390	0.49	5.538	0.740	0.27
	500	0.394	0.50	6.632	0.725	0.28
	600	0.392	0.49	6.920	0.712	0.28

6.4 Results

We consider several scenarios, showing the limitations of the reference-point-based specification of temporal misalignment and how different alignment sets best capture different synchronization requirements. First, Section 6.4.1 shows that in the synchronized scenario, the reference-point-based specification reports non-zero misalignment. Second, Section 6.4.2 shows that this also leads to overly pessimistic measures for the misalignment under a certain amount of jitter. Third, we consider the case of a uniform time shift in Section 6.4.3, showing the difference between the anchored and relative alignment sets.

For each scenario, we report the maximum misalignment Δ under each specification. To translate these into errors, we use the robustness estimates from the evaluation described in Section 6.3. In Table 1, we give the full results on the KITTI dataset with threshold $\Delta = 100, 200, 300, 400, 500, 600$ ms for all alignment sets and the reference-point-based specification, reporting the position error and F1 score for each case. For a given Δ, these results represent the expected error at that misalignment level. Hence, Table 1 shows the error corresponding to the maximum misalignment Δ for that scenario and specification.

We note that the rows of Table 1 are not directly comparable across specifications, as the meaning of Δ differs: for instance, $\Delta = 100$ ms for the reference-point specification means each sample is within 100 ms of a single reference point, while for an alignment set it means each constraint's misalignment is at most 100 ms.



(a) Ref-point. **(b)** $\mathcal{A}_{\text{anchored-late}}$. **(c)** $\mathcal{A}_{\text{anchored-inter}}$.

Figure 7 Visualization of the synchronized, LiDAR centered scenario. For each misalignment specification the misalignment is shown in green.

Table 2 Misalignment (in ms) and fusion error under perfect synchronization (best values bold). Only alignment sets matching the LiDAR configuration have zero misalignment. Under perfect synchronization, anchored and relative variants of the same configuration yield identical results.

	Misalign. (ms)		F1 Score (Mean)		Pos. Error Mean / Max (m)	
Specification	**Cent.**	**Late**	**Cent.**	**Late**	**Cent.**	**Late**
Reference-point	100	100	0.831	0.831	0.27 / 3.5	0.27 / 3.5
$\mathcal{A}_{\text{anchored-late}}$	100	**0**	0.825	**1**	0.25 / 3.3	**0** / **0**
$\mathcal{A}_{\text{anchored-inter}}$	**0**	100	**1**	0.823	**0** / **0**	0.25 / 3.5
$\mathcal{A}_{\text{relative-late}}$	100	**0**	0.801	**1**	0.32 / 3.5	**0** / **0**
$\mathcal{A}_{\text{relative-inter}}$	**0**	100	**1**	0.808	**0** / **0**	0.31 / 4.8

6.4.1 Synchronized

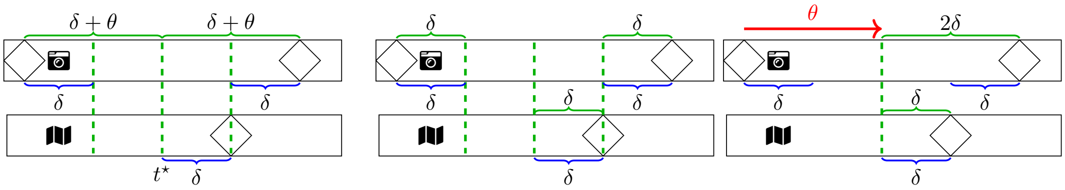
We first examine the baseline **synchronized** scenario described in Section 6.1, where the samples are taken exactly at their intended time points. That is, for an arbitrary beginning time point $t^\star$, the camera samples are taken at $t_{c_1} = t^\star$ and $t_{c_2} = t^\star + \theta$. The LiDAR sample is then taken at $t_{l_1} = t^\star + \frac{\theta}{2}$ for the LiDAR centered configuration and at $t_{l_1} = t^\star + \theta$ for the LiDAR late configuration. In this scenario, the system is behaving as intended, and a precise misalignment measure should report zero misalignment.

The left side of Table 2 shows the misalignment reported by different misalignment measures for both the LiDAR centered and LiDAR late scenarios. A visualization is shown in Figure 7. We only visualize the alignment sets $\mathcal{A}_{\text{anchored-inter}}$ and $\mathcal{A}_{\text{anchored-late}}$, the relative variants yield identical results under perfect synchronization, as shown in the table.

As shown in Table 2, the choice of misalignment specification is crucial for obtaining a precise measure of misalignment. The reference-point based misalignment measure fails to report zero misalignment. As this prior methods assumes that all samples should be aligned to a single reference point, it cannot distinguish between the intended offset and actual unintended misalignment.

In contrast, the alignment set approach is more expressive. By explicitly modeling that there is an intended offset, the alignment set that accurately matches the scenario configuration correctly reports zero misalignment. Selecting the correct alignment set is therefore essential. If an incorrect alignment set is selected, e.g., the one that assumes the LiDAR is centered when it is actually late, a non-zero misalignment of 100 ms is reported.

We apply the robustness estimates from Table 1 as described at the beginning of Section 6.4 to determine the error corresponding to the reported misalignment. Since the system is synchronized, one would expect zero error. However, the non-zero misalignments reported lead to overly pessimistic robustness evaluations as can be seen in the right side of Table 2.



(a) Ref-point. **(b)** $\mathcal{A}_{\text{anchored-inter}}$. **(c)** $\mathcal{A}_{\text{relative-inter}}$.

Figure 8 Visualization of the jitter scenario. Jitter is shown in blue, misalignment in green. For the anchored scenario the jitter is the same as the misalignment. For the relative scenario, the jitter can cause misalignment in either direction, leading to a maximum misalignment of $2j$.

Table 3 Reported misalignment and fusion error under jitter j for the LiDAR centered scenario. The matching alignment set ($\mathcal{A}_{\text{anchored-inter}}$) reports only the jitter, while other specifications include intended offsets or compound jitter, inflating both the reported misalignment and error bounds. Best values are in bold.

	Misalign. (ms)		F1 Score (Mean)		Pos. Error Mean / Max (m)	
Specification	$j = 100$	$j = 200$	$j = 100$	$j = 200$	$j = 100$	$j = 200$
Reference-point	200	300	0.764	0.722	0.36 / 4.5	0.38 / 4.6
$\mathcal{A}_{\text{anchored-inter}}$	**100**	**200**	**0.823**	**0.782**	**0.25** / **3.5**	**0.34** / **3.5**
$\mathcal{A}_{\text{anchored-late}}$	200	300	0.783	0.754	0.34 / **3.5**	0.37 / 4.6
$\mathcal{A}_{\text{relative-inter}}$	200	400	0.785	0.743	0.35 / 3.9	0.40 / 6.2
$\mathcal{A}_{\text{relative-late}}$	300	500	0.758	0.725	0.38 / 5.3	0.39 / 6.6

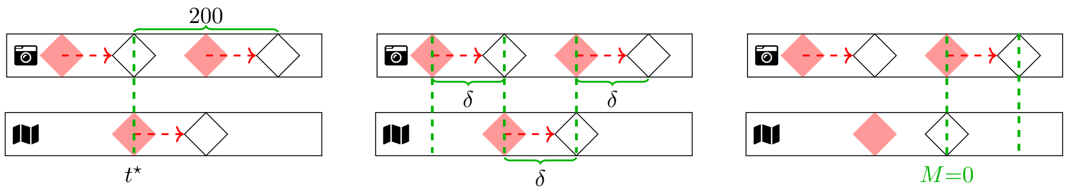
6.4.2 Jitter Misalignment

We now consider the **Jitter** scenario described in Section 6.1, to show how imprecise misalignment specification fail to separate jitter with the intended offset. In this scenario, sensor samples deviate from their intended time points by a random jitter offset in the range of $[-j, j]$ for some j. Ideally, a misalignment measure should isolate this jitter from the intended offset, reporting at most j as the misalignment. If a measure fails to distinguish between the intended offset and jitter, a much larger misalignment is reported, leading to overly pessimistic error bounds.

We determine the maximum misalignment that can be reported under a jitter of j ms for the different misalignment specification in the LiDAR centered configuration, showing the results on the left side of Table 3. The scenario is visualized in Figure 8.

Similar to the perfect synchronization scenario, the additional expressiveness of the alignment set approach is necessary to isolate the jitter. As shown in the table, only the correctly matching alignment set reports the jitter j as misalignment. The reference-point based specification reports a misalignment of $j+\theta$ ms, including the intended offset θ. Relative alignment sets count the jitter twice (either $2j$ for $\mathcal{A}_{\text{relative-inter}}$ or $2j + \theta$ for $\mathcal{A}_{\text{relative-late}}$), as jitter in either direction can cause misalignment for the relative constraints. The reason is that the second constraint references the first sample, whose jitter propagates into the computed misalignment of the second constraint, potentially doubling the reported jitter.

Imprecise misalignment measures report a larger misalignment Δ than the actual jitter j. This in turn causes the applicable robustness estimates to be overly pessimistic, as shown on the right side of Table 3. In all cases, using the correctly matching alignment set leads to the best error bounds. For the $j = 100$ ms case, the position error is around 40% worse for the reference-point based specification compared to the matching alignment set.



(a) Ref-point. **(b)** $\mathcal{A}_{\text{anchored-inter}}$. **(c)** $\mathcal{A}_{\text{relative-inter}}$.

Figure 9 Visualization of the uniform shift (red) with misalignment (green).

Table 4 Reported misalignment and fusion error under uniform time shifts of d for the centered scenario. The relative alignment set (**bold**) is invariant to the uniform shift, reporting zero misalignment and no error. The anchored alignment set (<u>underlined</u>) captures exactly the shift d, giving the best error bounds when uniform shifts should be considered misalignment. The prior specification give inflated misalignment measures and error bounds.

	Misalign. (ms)		**F1 Score (Mean)**		**Pos. Error Mean / Max (m)**	
Specification	d=100	d=200	d=100	d=200	d=100	d=200
Reference-point	200	300	0.764	0.722	0.36 / 4.5	0.38 / 4.6
$\mathcal{A}_{\text{anchored-inter}}$	<u>100</u>	<u>200</u>	0.823	0.782	0.25 / 3.5	0.34 / 3.5
$\mathcal{A}_{\text{anchored-late}}$	200	300	0.783	0.754	0.34 / 3.5	0.37 / 4.6
$\mathcal{A}_{\text{relative-inter}}$	**0**	**0**	**1**	**1**	**0 / 0**	**0 / 0**
$\mathcal{A}_{\text{relative-late}}$	100	100	0.801	0.801	0.32 / 3.5	0.32 / 3.5

6.4.3 Uniform Time Shift

We now consider the **Uniform shift** scenario described in Section 6.1, where all sensor samples are shifted by a uniform offset d, showing that alignment sets can capture different synchronization requirements for the same scenario. In practice, global clock drift can cause such uniform shifts, where the relative timing between samples is preserved but the samples are not taken at the intended time points.

Whether this uniform shift should be considered misalignment depends on the application's requirements. If only relative timing is important, then the shift should not be considered misalignment. If the fusion must correspond to specific moments, then the shift should be considered misalignment. Ideally, for this scenario the misalignment measure should therefore either report zero misalignment or a misalignment corresponding to the uniform shift d, depending on which of the two cases is considered.

The relative alignment sets $\mathcal{A}_{\text{relative-inter}}$ and $\mathcal{A}_{\text{relative-late}}$ are designed to capture the case where only relative timing is important, while the anchored alignment sets $\mathcal{A}_{\text{anchored-inter}}$ and $\mathcal{A}_{\text{anchored-late}}$ are designed to capture the case where specific time points are important.

The results for the LiDAR centered scenario are shown on the left side of Table 4, showing the misalignment reported for a uniform time shift of $d = 100$ ms and $d = 200$ ms.

The anchored alignment sets correctly capture only the uniform shift d as misalignment, while the relative alignment sets correctly report zero misalignment. In contrast, the reference-point-based misalignment is sensitive to the shift but adds the intended offset on top.

The robustness evaluation is shown on the right side of Table 4. The relative alignment set is invariant to the uniform shift, reporting zero misalignment and no error. If a uniform shift is not considered misalignment, and it is not an issue that the result represents a later time point, this is the best estimate. If the uniform shift is considered misalignment, the behavior is the same as the jitter case, with the anchored alignment set giving the best error.

Table 5 Mean F1 score and mean position error (m) per specification, with and without drift.

Specification	**Jitter Δ (ms)**	**F1 Score (Mean)** No drift	**F1 Score (Mean)** 10 ms / 30 s	**Pos. Error Mean (m)** No drift	**Pos. Error Mean (m)** 10 ms / 30 s
Reference-point	125	0.749	0.604	0.54	0.68
$\mathcal{A}_{\text{anchored-inter}}$	25	**0.830**	0.658	**0.12**	0.67
$\mathcal{A}_{\text{anchored-late}}$	125	0.657	0.528	0.52	0.65
$\mathcal{A}_{\text{relative-inter}}$	50	0.766	**0.765**	0.41	**0.41**
$\mathcal{A}_{\text{relative-late}}$	150	0.430	0.430	0.71	0.71

6.5 Full Drive Results

We evaluate alignment sets for a full trace on the longest KITTI drive (8.5 minutes, 5177 camera frames at 10 Hz, cameras downsampled to 5 Hz). Our aim is to show that alignment sets can be applied to a more complex scenario. We consider a 25 ms jitter and a potential 10 ms drift every 30 seconds. We use alignment sets in two manners: to give a robustness estimate for the entire drive akin to the previous fusion-level scenarios and to investigate the correlation between misalignment and fusion error.

To allow misalignments smaller than the 100 ms granularity, we resort to synthetic data. We interpolate camera frames using RIFE [13] and LiDAR bounding boxes by linear interpolation of their parameters, to achieve an effective 5 ms granularity. We verified by manual inspection that the interpolated data does not introduce significant artifacts.

We extend the alignment sets given in Section 6.2 to consider all fusions in a drive trace. For the anchored alignment sets, for each fusion i we add a $t_i^\star$ and the same per-sample constraints as in Section 6.2, anchored to $t_i^\star$. We do the same for reference-point-based specification. For the relative alignment sets, we add constraints between samples used in the same fusion as before, but also add a constraint between the first camera sample $c_{i,1} = (S_{\text{cam}}, 2i-1)$ of each fusion to the second camera sample $c_{i-1,2} = (S_{\text{cam}}, 2i)$ of the previous fusion, to capture the intended relative timing between fusions: $c_{i,1} \sim c_{i-1,2} + \theta$.

We consider two scenarios: 1) **Jitter**: samples independently deviate from their intended time point by up to ±25 ms in 5 ms steps, and 2) **Jitter + Drift**: the jitter for each sample is up to ±25 ms and every 30 seconds all following samples are shifted by 10 ms.

The first experiment extends the robustness evaluation to the entire drive trace. For each specification, we take Δ to be the worst-case misalignment under the jitter-only scenario as reported by the specification measure (column Δ in Table 5, compare Section 6.4.2). For anchored specifications, drift accumulates over time. Relative specifications only observe drift at the moment a drift step occurs: for fusions before or after the step the shift applies to every sample and cancels out. To model this, we modify our aggregation function g to increase the effective misalignment threshold of affected constraints. We sample 1000 traces of the entire drive for each scenario and specification.

Table 5 shows the results, mirroring the findings of the fusion-level jitter and the uniform shift scenario. The alignment set ($\mathcal{A}_{\text{anchored-inter}}$) gives the best error bounds, while the reference-point-based specification gives overly pessimistic estimates. The relative specifications are invariant to the uniform shift caused by drift, giving nearly identical results.

We also investigate how the misalignment measure $M_{\mathcal{A}}(T)$ correlates with the fusion error, by sampling 1000 traces for the scenarios from Experiment 1. Under each specification's alignment set, we compute the misalignment for each fusion and its fusion error.

We note that calculating the misalignment of 1000 traces takes less than a second, but evaluating the fusion error under an alignment set spans the entire drive, requiring a considerable computation cost of in total 20 hours for 1000 traces. A single trace consists of

Table 6 Per-spec correlation between misalignment $M_{\mathcal{A}}(T)$ and fusion position error. Best per column in bold.

Specification	No drift, E_{shifted}		No drift, $E_{\text{unshifted}}$		Drift 10/30s, E_{shifted}		Drift 10/30s, $E_{\text{unshifted}}$	
	r	ρ	r	ρ	r	ρ	r	ρ
Reference-point	+0.041	+0.051	+0.002	+0.001	+0.022	+0.026	+0.766	+0.782
$\mathcal{A}_{\text{anchored-inter}}$	+0.311	+0.282	**+0.369**	**+0.339**	+0.086	+0.089	**+0.811**	**+0.829**
$\mathcal{A}_{\text{anchored-late}}$	+0.010	+0.009	-0.003	-0.004	+0.027	+0.023	+0.597	+0.536
$\mathcal{A}_{\text{relative-inter}}$	**+0.386**	**+0.352**	+0.148	+0.155	**+0.384**	**+0.352**	-0.001	-0.001
$\mathcal{A}_{\text{relative-late}}$	+0.009	+0.006	+0.000	-0.001	+0.011	+0.009	-0.130	-0.138

1294 individual fusions, resulting in ≈ 1.3 million fusions in total. On average, calculating the error of one individual fusion takes about 0.05 seconds. The sampling takes less than 10 seconds with a single trace consuming less than 1 MB. Peak memory consumption evaluating one trace is 3.3 GB.

We use two fusion error functions, capturing different application requirements. $E_{\text{unshifted}}$ compares against the fusion of the original, unjittered and undrifted samples. For the unshifted error function, the drift causes the error to continuously increase, even though fusions are locally correct. E_{shifted} compares against the fusion of the samples at $(c_1, c_1 + \theta, c_1 + \frac{\theta}{2})$, where c_1 is the actual (jittered and drifted) first camera timestamp. The two error functions correspond respectively to the use cases captured by anchored and relative alignment sets.

Table 6 reports Pearson (r) and Spearman (ρ) correlations for each combination of specification, scenario and error function [30]. Pearson's $r = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$ measures linear correlation and Spearman's ρ monotonic correlation by applying Pearson to the ranks of X and Y. Both lie in in $[-1, 1]$, with 0 indicating no correlation. The results mirror the findings of the previous experiments. The correctly matching alignment set gives the best correlation with the error function corresponding to its use case.

7 Conclusion

We introduced *alignment sets*, a method for defining temporal misalignment that allows system designers to specify the intended alignment of a specific application. Robustness guarantees derived from alignment sets generalize definitions from prior work, while offering more precise misalignment measures in scenarios where samples are intentionally offset.

Our evaluation of a camera-LiDAR fusion for 3D object detection where two camera samples are fused with a single LiDAR sample shows that alignment sets yield more precise misalignment measures and tighter error bounds than prior work. Prior work assumes that all samples should be aligned to a common reference time point. This causes them to fail to capture the intended alignment of these scenarios, reporting non-zero misalignment even in the case of perfect synchronization and overestimating the misalignment in the case of jitter. In contrast, a well-designed alignment set is capable of expressing the requirements correctly. Alignment sets also allow specifying whether uniform shifts should be considered misalignment or not, depending on the application's requirements.

The choice of alignment set is crucial to obtain precise measures and guarantees. An alignment set designed for another scenario leads to overly pessimistic results. The flexibility of alignment sets allows to express the intended alignment of a wide range of scenarios, but also requires careful design to ensure that the intended alignment is correctly captured.

While alignment sets are expressive and flexible, how to correctly design them for more complex scenarios remains an open problem. Furthermore, we have limited our study to explore only linear alignment constraints. Generalization of alignment constraints, e.g., using nearest neighbor alignment, is an interesting direction for future work.

Comprehensive, large-scale evaluations require a considerable amount of time. In our camera-LiDAR fusion scenario it takes one hour to evaluate 77500 samples for one misalignment threshold under an alignment set covering one fusion. For the scenario of an alignment set covering a full drive, evaluating the fusion error for 1000 traces takes around 20 hours under one alignment set. Even larger alignment sets and more complex, expensive aggregation functions will further increase these costs. How to efficiently evaluate the error of sensor fusion due to temporal misalignment is also an interesting direction for future work. Our analysis of the misalignment measure shows that the time complexity depends on the cardinality of the alignment set and the cost of the aggregation function, ignoring the geometry of the constraints. An interesting future work is to derive tighter analysis of the time complexity or a more efficient misalignment measure by considering the geometry of the constraints and the aggregation function.

Not all alignment set constraints are expressible in plain STL, but extensions with parameters [2] and past operators [19] should be sufficient to express all constraints used in this work. Formalizing our alignment sets in such an extended STL and finding a tractable fragment is another interesting direction for future work.

References

1 Michael Aeberhard and Nico Kaempchen. High-level sensor data fusion architecture for vehicle surround environment perception. In *Proc. 8th Int. Workshop Intell. Transp*, volume 665, 2011.

2 Eugene Asarin, Alexandre Donzé, Oded Maler, and Dejan Nickovic. Parametric identification of temporal properties. In Sarfraz Khurshid and Koushik Sen, editors, *Runtime Verification – Second International Conference, RV 2011, San Francisco, CA, USA, September 27-30, 2011, Revised Selected Papers*, Lecture Notes in Computer Science, pages 147–160. Springer, 2011. doi:10.1007/978-3-642-29860-8_12.

3 Michal Bednarek, Piotr Kicki, and Krzysztof Walas. On robustness of multi-modal fusion—robotics perspective. *Electronics*, 9(7):1152, 2020.

4 Said Yacine Boulahia, Abdenour Amamra, Mohamed Ridha Madi, and Said Daikh. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Machine Vision and Applications*, 32(6):121, 2021. doi:10.1007/s00138-021-01249-8.

5 Varuna De Silva, Jamie Roche, and Ahmet Kondoz. Robust fusion of lidar and wide-angle camera data for autonomous mobile robots. *Sensors*, 18(8):2730, 2018. doi:10.3390/s18082730.

6 Alexandre Donzé. Breach, A toolbox for verification and parameter synthesis of hybrid systems. In Tayssir Touili, Byron Cook, and Paul B. Jackson, editors, *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, Lecture Notes in Computer Science, pages 167–170. Springer, 2010. doi:10.1007/978-3-642-14295-6_17.

7 Alexandre Donzé, Thomas Ferrère, and Oded Maler. Efficient robust monitoring for STL. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification – 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, Lecture Notes in Computer Science, pages 264–279. Springer, 2013. doi:10.1007/978-3-642-39799-8_19.

8 Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In Krishnendu Chatterjee and Thomas A. Henzinger, editors, *Formal Modeling and Analysis of Timed Systems – 8th International Conference, FORMATS 2010, Klosterneuburg, Austria, September 8-10, 2010. Proceedings*, Lecture Notes in Computer Science, pages 92–106. Springer, 2010. doi:10.1007/978-3-642-15297-9_9.

9 Georgios E. Fainekos and George J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theor. Comput. Sci.*, 410(42):4262–4291, 2009. doi:10.1016/j.tcs.2009.06.021.

10 Thomas Ferrere. Assertions and measurements for mixed-signal simulation. *Ph. D. Dissertation*, 2016.

11 Andreas Finkenzeller, Andrew Roberts, Mauro Bellone, Olaf Maennel, Mohammad Hamad, and Sebastian Steinhorst. Sensor fusion desynchronization attacks. In *37th Euromicro Conference on Real-Time Systems (ECRTS 2025)*, pages 6:1–6:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ECRTS.2025.6.

12 Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. doi:10.1177/0278364913491297.

13 Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022 – 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIV*, Lecture Notes in Computer Science, pages 624–642. Springer, 2022. doi:10.1007/978-3-031-19781-9_36.

14 Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE, 2018. doi:10.1109/ICCPS.2018.00035.

15 Taewan Kim and Joydeep Ghosh. On single source robustness in deep fusion models. *Advances in Neural Information Processing Systems*, 32, 2019.

16 Daniel Kuhse, Nils Hölscher, Mario Günzel, Harun Teper, Georg von der Brüggen, Jian-Jia Chen, and Ching-Chi Lin. Sync or sink? the robustness of sensor fusion against temporal misalignment. In *30th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2024, Hong Kong, May 13-16, 2024*, pages 122–134. IEEE, 2024. doi:10.1109/RTAS61025.2024.00018.

17 Elmar Mair, Michael Fleps, Michael Suppa, and Darius Burschka. Spatio-temporal initialization for imu to camera registration. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 557–564. IEEE, 2011. doi:10.1109/ROBIO.2011.6181345.

18 Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In Yassine Lakhnech and Sergio Yovine, editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004, Proceedings*, Lecture Notes in Computer Science, pages 152–166. Springer, 2004. doi:10.1007/978-3-540-30206-3_12.

19 Dejan Nickovic and Tomoya Yamaguchi. RTAMT: online robustness monitors from STL. In Dang Van Hung and Oleg Sokolsky, editors, *Automated Technology for Verification and Analysis – 18th International Symposium, ATVA 2020, Hanoi, Vietnam, October 19-23, 2020, Proceedings*, Lecture Notes in Computer Science, pages 564–571. Springer, 2020. doi:10.1007/978-3-030-59152-6_34.

20 Chanoh Park, Peyman Moghadam, Soohwan Kim, Sridha Sridharan, and Clinton Fookes. Spatiotemporal camera-lidar calibration: A targetless and structureless approach. *IEEE Robotics and Automation Letters*, 5(2):1556–1563, 2020. doi:10.1109/LRA.2020.2969164.

21 Won Park, Nan Liu, Qi Alfred Chen, and Z Morley Mao. Sensor adversarial traits: Analyzing robustness of 3d object detection sensor fusion models. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 484–488. IEEE, 2021. doi:10.1109/ICIP42928.2021.9506183.

22 Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE, 2018. doi:10.1109/IROS.2018.8593603.

23 Neha Rino, Mohammed Foughali, and Eugene Asarin. Efficiently computable distance-based robustness for a practical fragment of STL. In Jane Hillston, Sadegh Soudjani, and Masaki Waga, editors, *Quantitative Evaluation of Systems and Formal Modeling and Analysis of Timed Systems – First International Joint Conference, QEST+FORMATS 2024, Calgary, AB, Canada, September 9-13, 2024, Proceedings*, Lecture Notes in Computer Science, pages 179–195. Springer, 2024. doi:10.1007/978-3-031-68416-6_11.

24 Neha Rino, Mohammed Foughali, Florian Renkin, and Eugene Asarin. Efficiently computable temporal robustness for a practical stl fragment. *International Journal on Software Tools for Technology Transfer*, March 2026. doi:10.1007/s10009-026-00852-2.

25 Md Hasan Shahriar, Md Mohaimin Al Barat, Harshavardhan Sundar, Ning Zhang, Naren Ramakrishnan, Y Thomas Hou, and Wenjing Lou. Temporal misalignment attacks against multimodal perception in autonomous driving. *arXiv preprint*, 2025. arXiv:2507.09095.

26 Yde Sinnema and Martina Maggio. Analysis of control systems under sensor timing misalignments. In *2025 IEEE 31st Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 137–150. IEEE, 2025. doi:10.1109/RTAS65571.2025.00024.

27 James Tu, Huichen Li, Xinchen Yan, Mengye Ren, Yun Chen, Ming Liang, Eilyan Bitar, Ersin Yumer, and Raquel Urtasun. Exploring adversarial robustness of multi-sensor perception systems in self driving. *arXiv preprint*, 2021. arXiv:2101.06784.

28 Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. In *European conference on computer vision*, pages 1–21. Springer, 2024. doi:10.1007/978-3-031-72751-1_1.

29 Shaojie Wang, Tong Wu, Ayan Chakrabarti, and Yevgeniy Vorobeychik. Adversarial robustness of deep sensor fusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2387–2396, 2022.

30 Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010.

31 Antje Westenberger, Tobias Huck, Martin Fritzsche, Tilo Schwarz, and Klaus Dietmayer. Temporal synchronization in multi-sensor fusion for future driver assistance systems. In *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, pages 93–98. IEEE, 2011.

32 Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

A Proofs

Proof of Lemma 10. We establish the equivalence through proof by construction, giving an $\mathcal{A}, g$, and E such that the general robustness condition with these definitions is equivalent to the reference-point-based definition in Eq. 5.

We constrain every sample to $t^\star$: $\mathcal{A} = \{(S_i, 1) \sim t^\star \mid S_i \in \{S_1, ..., S_n\}\}$. As g we use the maximum of all absolute misalignments $g(m_1, \ldots, m_n) = \max_i |m_i|$. Substituting g and $\mathcal{A}$ into Definition 7, gives: $\max_{i \in \{1,\ldots,n\}} |t_i - t^\star| \leq \Delta$, equivalent to $t_i \in [t^\star - \Delta, t^\star + \Delta]$.

The error function uses the loss function: $E(T, f(T)) = L(f(t^\star, ..., t^\star), f(T))$. Substituting into Definition 9, gives $(\forall T, [M_{\mathcal{A}}(T) \leq \Delta] \Rightarrow E(T, f(T)) \leq \epsilon) \iff (\forall t_i \in [t^\star - \Delta, t^\star + \Delta] : L(f(t^\star, ..., t^\star), f(T)) \leq \epsilon)$ ◀

Proof of Lemma 11. We establish the equivalence through proof by construction, giving an $\mathcal{A}, g$, and E such that the general robustness condition with these definitions is equivalent to the global variants of the sample-based Definitions in Eq. 7 and Eq. 8.

We define A to align all samples to each other: $\mathcal{A} = \{(S_i, 1) \sim (S_j, 1) \mid S_i, S_j \in \{S_1, ..., S_n\}, i \neq j\}$. Using $g(m_1, \ldots, m_n) = \max_{i,j\in\{1,\ldots,n\}} |m_i - m_j|$ and substituting into Definition 7 gives $M_\mathcal{A} = \max_{i,j\in\{1,\ldots,n\}} |t_i - t_j|$. Limiting this to Δ limits the interval range $[\min(t_i), \max(t_i)]$ to Δ.

For strong sample-based robustness, with $T = (t_1, \ldots, t_n)$ we define:

$$E_{\text{strong}}(T, f(T)) = \max_{t^\star \in [\min(t_i), \max(t_i)]} L(f(T),\ f(t^\star, \ldots, t^\star)).$$

Substituting into Definition 9, gives $(\forall T, [M_\mathcal{A}(T) \leq \Delta] \Rightarrow E_{\text{strong}}(T, f(T)) \leq \epsilon)$
$\iff (\forall T, \max_{i,j} |t_i - t_j| \leq \Delta \Rightarrow \forall t^\star \in [\min(t_i), \max(t_i)],\ L(f(t^\star, \ldots, t^\star), f(T)) \leq \epsilon)$

For weak sample-based robustness, with $T = (t_1, \ldots, t_n)$ we define:

$$E_{\text{weak}}(T, f(T)) = \min_{t^\star \in [\min(t_i), \max(t_i)]} L(f(T),\ f(t^\star, \ldots, t^\star)).$$

Substituting into Definition 9: $(\forall T, [M_\mathcal{A}(T) \leq \Delta] \Rightarrow E_{\text{weak}}(T, f(T)) \leq \epsilon)$
$\iff (\forall T, \max_{i,j} |t_i - t_j| \leq \Delta \Rightarrow \exists t^\star \in [\min(t_i), \max(t_i)],\ L(f(t^\star, \ldots, t^\star), f(T)) \leq \epsilon)$ ◀