express it as a map, we will write it using standard notation for functions.

Definition 5 is the fundamental notion of behavioral substitutability that underlies our treatment in this paper. It is a non-trivial generalization of two prior concepts. One interesting and, as it turns out, particularly useful degree of flexibility is that the relation R can associate strings of differing lengths.

First, the preceding definition generalizes the earlier one:

Remark 3. When $\mathrm{R} = \mathbf{id}$, the identity relation, then Definition 5 recovers Definition 4 (the standard definition of output simulation, the subject of extensive prior study).

The specific requirement that F' handle at least the inputs that F does, explicit in Definition 4, becomes:

Property 6. For relation $\mathrm{R} \subseteq A \times B$, a necessary condition for any F to be R-simulatable is that $\mathrm{R} \cap (\mathscr{L}(F) \times B)$ be *left-total* in the sense that for every $s \in \mathscr{L}(F)$, there exist some t with $s\,\mathrm{R}\,t$. (Were it otherwise for some $s \in \mathscr{L}(F)$, then that string s suffices to violate condition 1 in Definition 5.)

And second, for the other generalization:

Remark 4. Definition 5 subsumes the ideas of *sensor maps* [31] (also called label maps). These are functions, $h : Y \to X$, taking individual observation symbols to another set. (One may model sensor non-ideality by applying such functions; for instance, observations $y \in Y$ and $y' \in Y$ can be conflated when $h(y) = h(y')$.) Specifically, sensor maps only give relations R restricted so that any $s\,\mathrm{R}\,t$ must have $|s| = |t|$, that is, the strings will have equal length.

Given a sensori-computational device F and general relation R, the question is whether any sensori-computational device F' exists to output simulate F modulo R. For the particular case that R is $\mathbf{id}$, F always output simulates itself. This fact means that the prior work focusing on minimizing filters, such as [28], [32], [26], can be reinterpreted as optimizing size subject to output simulation modulo $\mathbf{id}$.

Returning to more general relations R, the existence of a suitable F' is the central question in prior work on the destructiveness of label maps [31], [10]. General relations make the picture more complex, however, and these will be our focus as well as complex relations composed from more basic ones.

When R is many-to-1 it models compression or conflation. Output simulation of F modulo such a relation shows that F's behavior is unaffected by reduction of observation fidelity, i.e., it is compression that is functionally lossless. Relations R that are 1-to-many model noise via non-determinism. Output simulation modulo such relations show that operation is preserved under the injected uncertainty. And many-to-many relations treat both aspects simultaneously.

Going forward, we will use the open semi-colon symbol to denote relation composition, i.e., $\mathrm{U} \fatsemi \mathrm{V}$ is $\{(u,v) \mid \exists r \text{ s.t.} (u,r) \in \mathrm{U} \text{ and } (r,v) \in \mathrm{V}\}$. Beware that when both relations are functions (cf. Remark 2), the notation unfortunately reverses the convention for function composition, so $g \fatsemi f = f(g(\cdot)) = f \circ g$. (This will arise in, for example, Problem 2.)

Property 7. Given sensori-computational devices F and G, and left-total relations U and V on $\mathscr{L}(F) \times \mathscr{L}(G)$, with $\mathrm{U} \supseteq \mathrm{V}$, then $G \sim F \pmod{\mathrm{U}} \implies G \sim F \pmod{\mathrm{V}}$.

Hence, sub-relations formed by dropping certain elements do not cause a violation in output simulation if left-totalness is preserved. Before composing chains of relations, we examine further the connection raised in Remark 4.

Remark 5. Unlike sensor maps, the property of output simulating modulo some relation is not monotone under composition. For sensor maps, there is a notion of irreversible destructiveness: composition of a destructive map with any others is permanent, always resulting in a destructive map. That theory can talk meaningfully of a feasibility boundary in the lattice (e.g., title of [10]). For relations, composing additional relations can 'rescue' the situation. For instance, consider the device F_{rgb} in Figure 1. For relation $\mathrm{U} = \{(a,p),(a,q),(b,q),(b,t)\}$ there can be no G that output simulates F modulo U because q must either be green or blue, but can't be both. Formally $\{\text{green}\} = \mathscr{C}(F,a) \supseteq \mathscr{C}(G,q)$ since $a\,\mathrm{U}\,q$, and $\{\text{blue}\} = \mathscr{C}(F,b) \supseteq \mathscr{C}(G,q)$ since $b\,\mathrm{U}\,q$, and $\mathscr{C}(G,q) \neq \varnothing$. But with $\mathrm{V} = \{(p,a),(p,a'),(t,b),(t,b')\}$, which is not left-total (cf. Property 6), crucially, then it is easy to give some G' so that $G' \sim F_{\mathrm{rgb}} \pmod{\mathrm{U} \fatsemi \mathrm{V}}$. One can simply take F_{rgb} and add a' and b' to the edge sets with a and b, respectively.

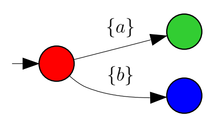


Fig. 1: A small sensori-computational device F_{rgb}, with $Y(F_{\mathrm{rgb}}) = \{a,b\}$, and $C = \{\text{red}, \text{green}, \text{blue}\}$.

Nevertheless, one may form a chain of relations:

Theorem 8. Given two relations R_1 and R_2, and sensori-computational device F, if there exists a sensori-computational device F_1 with $F_1 \sim F \pmod{\mathrm{R}_1}$ and there exists a sensori-computational device F_2 with $F_2 \sim F_1 \pmod{\mathrm{R}_2}$, then $F_2 \sim F \pmod{\mathrm{R}_1 \fatsemi \mathrm{R}_2}$.

Proof: Suppose that $\mathrm{R}_1 \subseteq A \times C$ and $\mathrm{R}_2 \subseteq D \times B$, then since $\mathscr{L}(F) \subseteq A$ and $B \subseteq \mathscr{L}(F_2)$, to establish that $F_2 \sim F \pmod{\mathrm{R}_1 \fatsemi \mathrm{R}_2}$, we verify the two required properties: 1) For all $s \in \mathscr{L}(F)$, $\exists u \in \mathscr{L}(F_1)$ with $s\,\mathrm{R}_1\,u$, and since $u \in \mathscr{L}(F_1)$, $\exists v \in \mathscr{L}(F_2)$ with $u\,\mathrm{R}_2\,v$. But then $(s,v) \in \mathrm{R}_1 \fatsemi \mathrm{R}_2$. 2) For any $(s,v) \in \mathrm{R}_1 \fatsemi \mathrm{R}_2$, there exists some $t \in C \cap D$ such that $s\,\mathrm{R}_1\,t$ and $t\,\mathrm{R}_2\,v$. Since $C \subseteq \mathscr{L}(F_1) \subseteq D$, the bridging $t \in \mathscr{L}(F_1)$. When this pair (s,v) has $s \in \mathscr{L}(F)$ then first: $\mathscr{C}(F,s) \supseteq \mathscr{C}(F_1,t)$ because $F_1 \sim F \pmod{\mathrm{R}_1}$; second: $\mathscr{C}(F_1,t) \supseteq \mathscr{C}(F_2,v)$ because $F_2 \sim F_1 \pmod{\mathrm{R}_2}$. Hence, $\mathscr{C}(F,s) \supseteq \mathscr{C}(F_1,t) \supseteq \mathscr{C}(F_2,v)$, as required. ∎

Since Sections IV and V will consider particular relations that model properties specifically related to event sensors, this

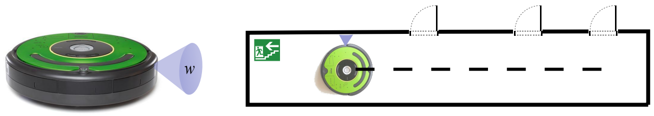


Fig. 2: An iRobot Create drives down a corridor its wall sensor w generating output values as it proceeds.

theorem can be useful when one is interested in devices under the composition of those relations.

IV. Structured observations: observation differencing

The most obvious fact about event cameras is that the phenomena they are susceptible to (photons) impinge on some hardware apparatus (the silicon retina) in a way which produces signals (intensity) for which differencing is a meaningful operation. We talk about 'events' as changes in those signals because we can define and identify differences (e.g., in brightness). Thus far, our formalized signal readings are only understood to involve elements drawn from $Y(F)$, just a set. The idea in this section is to contemplate structure in the raw signal space that permits some sort of differencing. Accordingly, the pair definitions that follow next.

Definition 9 (observation variator)**.** An *observation variator*, or just variator, for a set of observations Y is a set D and a ternary relation $\mathrm{S}_D \subseteq Y \times D \times Y$.

Often the two will be paired: (D, S_D). Reducing cumbersomeness, the S_D will be dropped sometimes, but understood to be associated with D and $Y(F)$ for some sensori-computational device F. Anticipating some cases later, when $(y,d,y') \in \mathrm{S}_D$ we may also write it as a function: $y' = \mathrm{S}_D(y,d)$. But beware of the fact that it may be multi-valued, and it may be partial.

On occasion we will call D the set of *differences*, terminology which aids in interpretation but should be thought of abstractly (as nothing ordinal or numerical has been assumed about either the sets $Y(F)$ or D).

Definition 10 (delta relation)**.** For a sensori-computational device F with variator (D, S_D), the associated *delta relation* is $\nabla\big|_F^D \subseteq \mathscr{L}(F) \times (\{\varepsilon\} \cup (Y(F) \cdot D^*))$ defined as follows:

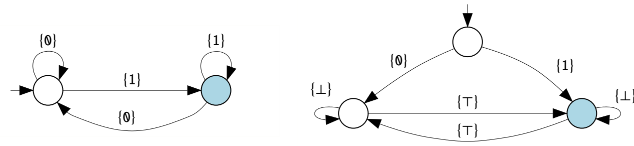
0) $\varepsilon \nabla\big|_F^D \varepsilon$, and
1) $y_0 \nabla\big|_F^D y_0$, for all $y_0 \in Y(F) \cap \mathscr{L}(F)$, and
2) $y_0 y_1 \ldots y_m \nabla\big|_F^D y_0 d_1 d_2 \ldots d_m$, where $(y_{k-1}, d_k, y_k) \in \mathrm{S}_D$.

Intuitively, the interpretation is that S_D tells us that d_k represents a shift taking place to get to y_k from symbol y_{k-1}. (With mnemonic 'difference' for d_k.)

Leading immediately to the following question:

Question 1. For any sensori-computational device F with variator (D, S_D), is it $\nabla\big|_F^D$-simulatable?

Given F with variator (D, S_D), we call a device F' that output simulates F modulo $\nabla\big|_F^D$ a *derivative* of F. In such cases we will say F has a derivative under the observation variator D.



(a) A sensori-computational device F_{wall}, with $Y(F_{\text{wall}}) = \{\texttt{0}, \texttt{1}\}$; within the vertices, white encodes $\{\texttt{0}\}$, and azure $\{\texttt{1}\}$.

(b) A sensori-computational device F'_{wall} with observation variator is $D_2 = \{\bot, \top\}$, so $F'_{\text{wall}} \sim F_{\text{wall}} \left(\bmod \nabla\big|_{F_{\text{wall}}}^{D_2}\right)$.

Fig. 3: Two sensori-computational devices describing the scenario in Example 1: (a) A model of the rather trivial transduction of the Create's wall sensor, and (b) its derivative under the observation variator D_2.

Example 1 (iRobot Create wall sensor)**.** In Figure 2 an iRobot Create moves through an environment. As it does this, the infrared wall sensor on its port side generates a series of readings. These readings (obtained via Sensor Packet ID: #8, with '$\texttt{0}$ = no wall, $\texttt{1}$ = wall seen' [12, pg. 22]) have binary values. The Create's underlying hardware realizes some basic computation on the raw sensor to produce these values by thresholding luminance, either as a voltage comparison via analogue circuitry or after digital encoding. To cross the hardware/software interface, the detector's binary signal is passed through a transducer, namely sensori-computational device of the form shown in Figure 3a.

A suitable observation variator is $D_2 = \{\bot, \top\}$, and ternary relation written in the form of a table as $(\text{row}, \text{cell-entry}, \text{column}) \in \mathrm{S}_{D_2} \subseteq \{\texttt{0}, \texttt{1}\} \times D_2 \times \{\texttt{0}, \texttt{1}\}$ as:

S_{D_2}	0	1
0	$\bot$	$\top$
1	$\top$	$\bot$

Using this variator, there is a sensori-computational device that output simulates F_{wall} modulo the delta relation $\nabla\big|_{F_{\text{wall}}}^{D_2}$. Figure 3b shows its derivative F'_{wall}. A direct interpretation for how D_2 encodes the variation in the bump signal is that $\top$ indicates a flip in the signal; while $\bot$ makes no change. Also, the first item in the sequence, some element from $Y(F_{\text{wall}})$, describes the offset from wall at time of initialization. $\square$

The preceding example, though simple, illustrates why we have started from the very outset by considering stateful devices. This may have seemed somewhat peculiar because we are treating sensors and these are seldom conceived of as especially stateful. An event sensor requires *some* memory, and so state is a first-class part of the model. (As already touched upon, some authors have applied the moniker 'virtual' to sensors that involve some computational processing.)

Especially when exploring aspects of the delta relation's definition, most of our examples will involve very simple input–output mappings. It should be clear that they could quickly become rather more complex. For instance if, in

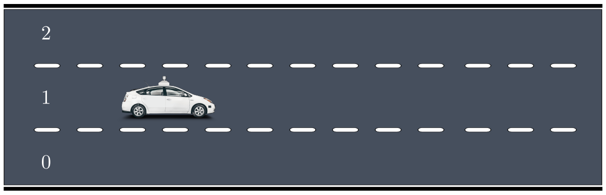


Fig. 4: A self-driving car, with on-board sensors to detect whether the vehicle changes to the left or right, or stays at the current lane.

Figure 2, the robot must tell apart odd and even doors, then a suitable adaption of the sensor is easy to imagine: the 2 states in Figure 3a become 4, and the outputs involve three colors, *etc*.

Example 2 (Lane sensor)**.** A self-driving car, shown in Figure 4, moves on a highway with three lanes. It is equipped with on-board LiDAR sensors to detect the vehicle's current lane. Supposing these are indexed from its right to left as 0, 1 and 2, then this lane sensor produces one of these three outputs. To construct a sensor that reports a change in the current lane, consider the observation variator is $(D_{\text{3-lane}}, \mathrm{S}_{D_{\text{3-lane}}})$ with $D_{\text{3-lane}} = \{\textsc{left}, \textsc{null}, \textsc{right}\}$ and, $\mathrm{S}_{D_{\text{3-lane}}}(i,d) = \min(\max(i+v(d)),0),2)$, where $v(\textsc{left}) = +1$, $v(\textsc{null}) = 0$, and $v(\textsc{right}) = -1$.

This observation variator will be able to transform any sequence of lane occupations into unique lane-change signals in a 3-lane road. □

Example 3 (Minispot with a compass)**.** Consider a Minispot, the Boston Dynamics quadruped robot in Figure 5. Assume that it is equipped with motion primitives that, when activated, execute a gait cycle allowing it to move forward a step, move backward a step, or turn in place $\pm 45°$, without losing its footing. Starting facing North, after each motion primitive terminates, the Minispot's heading will be one of 8 directions (the 4 cardinal plus 4 intercardinal ones).

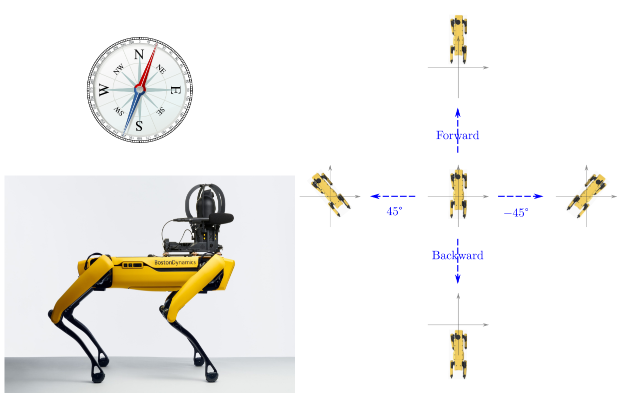


Fig. 5: A Boston Dynamics Minispot quadraped is equipped with a compass to give its heading. To simplify the control challenge, the Minispot is programmed with motion primitives to step forward and backward, and to turn in place by $\pm 45°$ (illustrated on the right).

Suppose the raw compass produces measurements $x \in \{\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow\}$, then consider the observation variator (D_3, S_{D_3}) with $D_3 = \{-, \emptyset, +\}$, where $(x, \emptyset, x) \in \mathrm{S}_{D_3}$, and $(x,+,y) \in \mathrm{S}_{D_3}$ if the angle from x to y is $45°$, and $(x,-,y) \in \mathrm{S}_{D_3}$ if the angle from x to y is $-45°$.

Given the four motion primitives, any sequence of those actions produces a sequence of compass measurements that the output variator D_3 can model. The constraint implied by such sequences means that a model of the raw compass will have a derivative under variator D_3. □

The previous two examples show that constraints can be imposed on the signal space—in the first case owing to the range of lanes possible (i.e., a saturation that arises); or via structure inherited from the control system (i.e., only some transitions are achievable). For both situations, a simple 3 element set is sufficient only because the sequences of changes the sensor might encounter has been limited.

Example 4 (Minispot with a compass, revisited)**.** Suppose the Minispot of Example 3 has been enhanced by supplementing its motion library with a primitive that allows it to turn in place by $\pm 90°$. Now, after each motion primitive terminates, the compass signal can include changes for which (D_3, S_{D_3}) is inadequate. When Definition 10 is followed to define $\nabla\big|_F^{D_3}$, those sequences involving $90°$ changes fail to find any $d_k \in D_3$, and the relation is not left-total. Hence, via Property 6, there can be no derivative.

Naturally, a more sophisticated observation variator does allow a derivative. Supposing we encode $\{\uparrow, \nearrow, \ldots \nwarrow\}$ instead with headings as integers $\{0, 45, \ldots, 315\}$, then we might consider the variator $(\mathbb{Z}, +)$. By the plus we refer to the function $+ \colon \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$, the usual addition on integers. To meet the requirements of Definition 9 strictly, we ought to take the restriction to the subset of triples (in the relation) where the first and third slots only have elements within $\{0, 45, \ldots, 315\}$. Then there are devices that will output simulate modulo $\nabla\big|_F^{\mathbb{Z}}$. □

The preceding illustrates how, for a sufficient observation variator, (D, S_D), some aspect of the signal's variability is expressed in the cardinality of D. Thus, seeking a small (or the smallest possible) D will be instructive; employing the whole kitchen sink, as was done with the integers above, fails to pinpoint the necessary information. Also, having stated that some variators are sufficient and touched upon Property 6, the following remark is in order.

Remark 6. Despite Property 6, one does not require that S_D be left-total for Question 1 to have an affirmative answer. For instance, some pairs of y and y' may never appear sequentially in strings in $\mathscr{L}(F)$, and so S_D needn't have any triples with y and y' together. (Though, obviously, the left-totalness of $\nabla\big|_F^D$ is required.)

Proposition 11. A sufficient condition for an affirmative answer to Question 1 is that $y' = \mathrm{S}_D(y,d)$ be a single-valued partial function whose $\nabla\big|_F^D$ is left-total. More explicitly, the