Spatio-Temporal Reasoning about Traffic Scenarios

Christoph Schwering and **Gerhard Lakemeyer**
Knowledge-based Systems Group
RWTH Aachen University, Aachen, Germany
(schwering,gerhard)@kbsg.rwth-aachen.de

Abstract

A model of spatial relations between automobiles changing over time needs to trade off expressivity, computational complexity, type of measured data, and closeness to human cognition. We present a common sense theory for reasoning about distances between cars based on two temporal distances between pairs of cars, *net time gap* and *time to collision*. We give an axiomatization in the situation calculus which allows reasoning about car-to-car relations and how they are affected by time and acceleration. We also discuss experimental results in a plan recognition scenario.

Introduction

When modeling spatio-temporal relations of cars one needs to walk a fine line between realism and a suitable level of abstraction. Since automobile driving is a physically complex process, realistic models are often also complex, both mathematically and computationally (Helbing and Nagel 2004). For some applications, however, common sense oriented models seem more suitable, e.g., for plan recognition. The goal of plan recognition is to infer from a pre-defined plan library which plan an observed agent pursues. Simple models are desirable in this setting to keep the process of creating the plan library manageable and its result clearer. Also, online plan recognition requires fast reasoning in the model in order to quickly process new information.

In (Schwering et al. 2012) we presented a model-based approach to plan recognition. The plan library consists of programs, whose execution is simulated. The result of this simulation is compared to observations of the real world. In the traffic domain this means that the simulated vehicles' position etc. is compared to the real position. Thus we can determine those programs from the plan library which match the observed driving maneuver.

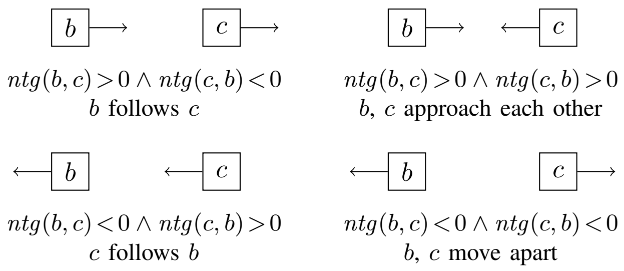
For lateral movement, lane markings act as landmarks by which drivers orient themselves. When it comes to longitudinal movement, there is no such global reference point. Instead, this role is filled by the other dynamically moving vehicles on the road. In this paper we present a model to express this feature. Our theory is based on two temporal distance measures to describe the longitudinal relation between two vehicles: The *net time gap* (NTG) between two cars denotes the time that passes until one car has reached the position of the other, i.e., it measures how close two cars are. The *time to collision* (TTC) is the time it takes one car to catch up with the other one, i.e., it measures how fast cars are approaching one another. These temporal measures have the advantage of being less dependent on cars' velocities when it comes to assessing the hazard potential of a situation. Additionally they are relatively easy to measure in reality and actually used in practice (Marsden, McDonald, and Brackstone 2001; Dagli, Brost, and Breuel 2002). We show that from a chain of NTG and TTC measurements of adjacent cars the NTG and TTC relations of all cars can be inferred. This gives rise to a view of the entire traffic situation. We also integrate the concepts of time and acceleration in this model to account for the dynamics in the traffic domain.

Since, as we will see, it turns out to be useful to work with actual quantitative measurements in traffic scenarios, purely qualitative calculi are not suitable for our purposes. Calculi like Allen's Algebra (Allen 1983), the Point Algebra (Vilain, Kautz, and Beek 1986), the Region Connected Calculus (RCC) (Randell, Cui, and Cohn 1992), or the Oriented Point Algebra (OPRA) (Mossakowski and Moratz 2012) focus more on the representation of different constellations of spatial or temporal relations, less on their temporal evolution. They provide no means to reason about closeness and the rate at which it changes, i.e., the speed at which cars approach each other. (Clementini, Di Felice, and Hernández 1997) does consider distance, but it also lacks a concept of time. In fast-paced scenarios like traffic, however, this temporal evolution is of great importance. While the Qualitative Trajectory Calculus (QTC) (Van de Weghe et al. 2006; 2005) has been applied to traffic scenarios, it seems to be inappropriate to assess the hazard potential of traffic situations because it reduces the continuum to three values, $-$, 0, $+$, and thus loses a representation of time as needed for fast-paced situations as they occur in traffic.

The rest of the paper is organized as follows. In the next section we present a model for longitudinal reasoning using the measures net time gap and time to collision. The subsequent section shows how we make use of this model for plan recognition. In Section 4, we present experimental results in this application. Then we conclude.

Model

Our model is based on two temporal distance measures to describe the longitudinal relation between two vehicles: The



$ntg(b,c) > 0 \wedge ntg(c,b) < 0$
b follows c

$ntg(b,c) > 0 \wedge ntg(c,b) > 0$
b, c approach each other

$ntg(b,c) < 0 \wedge ntg(c,b) > 0$
c follows b

$ntg(b,c) < 0 \wedge ntg(c,b) < 0$
b, c move apart

Figure 1: The sign of NTG indicates the driving direction: $ntg(b,c)$ is positive iff b moves towards c's position.

net time gap (NTG) between two cars b and c denotes the time (usually in seconds) that passes until b has reached the position at which c is right now. The *time to collision* (TTC) denotes the time that passes until b has caught up with c.

First we define NTG and TTC in the context of a global view. Then we show what can be inferred from NTG and TTC alone without any global information. Let $x(b)$ be the global longitudinal position of a car b (i.e., the distance to some imaginary starting line) and $v(b)$ its longitudinal velocity. Then NTG and TTC are determined by

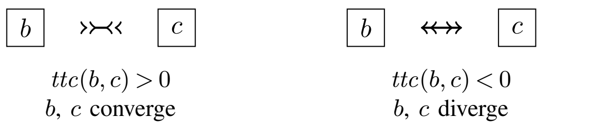
$$ntg(b,c) = \tfrac{x(c)-x(b)}{v(b)} \quad \text{and} \quad ttc(b,c) = \tfrac{x(c)-x(b)}{v(b)-v(c)}$$

Intuitively NTG measures how close b is to c. The two-second rule, a common guideline to keep a safe distance to the vehicle in front,[1] simply says to keep $ntg(b,c) \geq 2$. Besides the absolute value, the sign of NTG is interesting, because it tells us in which direction a car moves: $ntg(b,c) > 0$ means that b moves towards c's current position, i.e., b will be at c's position in $ntg(b,c)$ seconds. On the other hand $ntg(b,c) < 0$ means b moves away from c's position, i.e., b has been at c's position $-ntg(b,c)$ seconds ago. Figure 1 depicts the possible NTG relations of b and c.

In traffic not just the closeness and driving direction are important, but also the rate at and direction in which closeness, i.e., the NTG, changes. This is captured by the TTC. When it is positive, both cars are actually getting closer. Otherwise the cars are diverging. These possible relations are depicted in Figure 2. Thus NTG and TTC together provide much information: they tell us the relative driving directions of both cars and how this relation will evolve over time.

When $v(b) = 0$ or $v(b) = v(c)$ then NTG and TTC are undefined, respectively, due to division by zero. This makes sense, because a still standing car will never reach another car's position, and two equally fast cars will never catch up.

In the rest of this section we will at first encode NTG and TTC in logic. The following subsections show that NTG and TTC are both symmetric and transitive when both NTG and TTC are known. By symmetry we mean that the perspective of NTG and TTC can be inverted. By transitivity we mean that from measurements of adjacent cars b, c and c, d we can compute the NTG and TTC of b and d. Then we show how time and acceleration affect NTG and TTC. Finally we present a situation calculus theory based on these results.



$ttc(b,c) > 0$
b, c converge

$ttc(b,c) < 0$
b, c diverge

Figure 2: The sign of TTC indicates whether b and c are getting closer or diverging from another.

Let $\mathcal{E}$ denote the following set which axiomatizes our model in the context of a global view:

$$\begin{aligned}\mathcal{E} = \{&NTG(b,c,r) \equiv v(b) \neq 0 \wedge r = \tfrac{x(c)-x(b)}{v(b)},\\ &TTC(b,c,r) \equiv v(b) \neq v(c) \wedge r = \tfrac{x(c)-x(b)}{v(b)-v(c)},\\ &\text{Axiomatization of real numbers}\}\end{aligned}$$

Free variables are implicitly universally quantified except when used in macro definitions. We use two sorts: *car* for traffic participants and *real* for the real numbers. We shall use b, c, d for variables of sort *car*, all other variables are of sort *real*. NTG and TTC are of sort $car \times car \times real$ while x and v are of sort $car \to real$. The axioms express that NTG and TTC are *partial functions*, i.e., their result r is uniquely determined by b and c if it is defined.

For brevity we often use a functional notation like $ntg(b,c)$. This notation is defined as follows: When an atomic sentence $P(t_1,\ldots,t_n)$ or $t_1 = t_2$ for some predicate symbol P and terms $t_1,\ldots,t_n$ mentions a term $ntg(b,c)$, this is a shorthand for $(\exists r)(NTG(b,c,r) \wedge P(t'_1,\ldots,t'_n))$ or $(\exists r)(NTG(b,c,r) \wedge t'_1 = t'_2)$, respectively, where t'_i results from simultaneously replacing all occurrences of $ntg(b,c)$ in t_i by a new variable r. Thus this expanded formula evaluates to False if NTG is not defined for b, c. Analogously proceed for TTC and also for the real number division.

We will also use the following abbreviations to assert that NTG and TTC are defined or non-zero, respectively:

$$\kappa(b,c) \overset{\mathsf{def}}{=} (\exists r)NTG(b,c,r) \wedge (\exists r)TTC(b,c,r) \quad \text{and}$$

$$\eta(b,c) \overset{\mathsf{def}}{=} \kappa(b,c) \wedge ntg(b,c) \neq 0 \wedge ttc(b,c) \neq 0$$

Combined knowledge of NTG *and* TTC entails several properties which do not hold if either of them is not known. E.g., the speed ratio of two cars b, c can be determined using only NTG and TTC without any further information:

Lemma 1

$$\mathcal{E} \models \eta(b,c) \supset \tfrac{v(c)}{v(b)} = 1 - \tfrac{ntg(b,c)}{ttc(b,c)}$$

Symmetry

In the following we show how to flip the perspective between two cars. That is, knowing $ntg(b,c)$ and $ttc(b,c)$ also implies knowledge about $ntg(c,b)$ and $ttc(c,b)$.

Proposition 2 (Symmetry of NTG)

$$\mathcal{E} \models \eta(b,c) \wedge ntg(b,c) \neq ttc(b,c) \supset$$
$$ntg(c,b) = \frac{-1}{1 - \frac{ntg(b,c)}{ttc(b,c)}} \cdot ntg(b,c)$$

[1] http://en.wikipedia.org/wiki/Two-second_rule

The condition $ntg(b,c) \neq ttc(b,c)$ covers division by zero in the consequent. It is violated if $v(c) = 0$.

Proposition 3 (Symmetry of TTC)

$$\mathcal{E} \models \kappa(b,c) \supset ttc(c,b) = ttc(b,c)$$

The first result follows easily from Lemma 1, the second follows directly from the definition of TTC.

Transitivity

Here we show that NTG and TTC are transitive. That is, when we know $ntg(b,c)$, $ntg(c,d)$ and $ttc(b,c)$, $ttc(c,d)$ we can compute $ntg(b,d)$ and $ttc(b,d)$.

Proposition 4 (Transitivity of NTG)

$$\mathcal{E} \models \eta(b,c) \wedge \kappa(c,d) \supset \\ ntg(b,d) = ntg(b,c) + \big(1 - \tfrac{ntg(b,c)}{ttc(b,c)}\big) \cdot ntg(c,d)$$

This can again be easily proven using Lemma 1.

Before showing transitivity of TTC we introduce the following lemma which helps to deal with relative velocities $v(b) - v(c)$. When the velocity of b (or c) changes by factor q to $q \cdot v(b)$ (or $q \cdot v(c)$), the new relative velocity can be written as $\alpha \cdot (v(b) - v(c))$ for some α which involves no functions other than NTG and TTC.

Lemma 5 (i) When b's velocity changes by factor q, the relative velocity $v(b) - v(c)$ changes to

$$\mathcal{E} \models \eta(b,c) \supset q \cdot v(b) - v(c) = g(b,c,q) \cdot \big(v(b) - v(c)\big)$$
$$\text{where } g(b,c,q) \overset{\text{def}}{=} (q-1) \cdot \tfrac{ttc(b,c)}{ntg(b,c)} + 1$$

(ii) When c's velocity changes by factor q, it is

$$\mathcal{E} \models \eta(b,c) \supset v(b) - q \cdot v(c) = h(b,c,q) \cdot \big(v(b) - v(c)\big)$$
$$\text{where } h(b,c,q) \overset{\text{def}}{=} (1-q) \cdot \tfrac{ttc(b,c)}{ntg(b,c)} + q$$

Proof. (i) With Lemma 1 we have:

$$\mathcal{E} \models \eta(b,c) \wedge q \neq 0 \supset \tfrac{v(c)}{q \cdot v(b)} = \tfrac{1}{q} \cdot \tfrac{v(c)}{v(b)} = \tfrac{1}{q} - \tfrac{1}{q} \cdot \tfrac{ntg(b,c)}{ttc(b,c)}$$
$$\mathcal{E} \models \eta(b,c) \wedge q \neq 0 \supset \tfrac{v(c)}{q \cdot v(b)} = 1 - \tfrac{g(b,c,q)}{q} \cdot \tfrac{ntg(b,c)}{ttc(b,c)}$$

This equation can be solved for $g(b,c,q)$.

(ii) $h(b,c,q)$ can be determined analogously. □

From this lemma the transitivity of TTC follows:

Proposition 6 (Transitivity of TTC)

$$\mathcal{E} \models \eta(b,c) \wedge \eta(c,d) \wedge \tfrac{ttc(b,c)}{ntg(b,c)} + \tfrac{ttc(c,d)}{ntg(c,d)} \neq 1 \supset \\ ttc(b,d) = \lambda_1 \cdot ttc(b,c) + \lambda_2 \cdot ttc(c,d)$$

where

$$\lambda_1 \overset{\text{def}}{=} \frac{ttc(c,d) \cdot ntg(b,c)}{ntg(c,d) \cdot ttc(b,c) + ttc(c,d) \cdot ntg(b,c) - ntg(c,d) \cdot ntg(b,c)}$$
$$\lambda_2 \overset{\text{def}}{=} \frac{ttc(b,c) \cdot ntg(c,d) - ntg(b,c) \cdot ntg(c,d)}{ntg(b,c) \cdot ttc(c,d) + ttc(b,c) \cdot ntg(c,d) - ntg(b,c) \cdot ntg(c,d)}$$

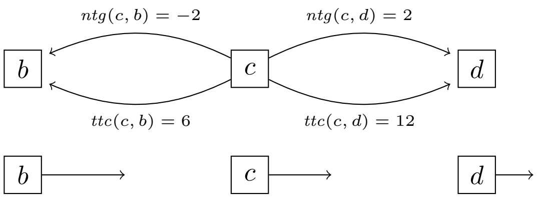


Figure 3: From the NTG and TTC measurements depicted in the top, we can infer the order and relative relations as shown in the bottom.

With these results we can compute a fairly complete view of the traffic situation from relatively little data. Consider Figure 3, in which we have only measured NTG and TTC between c and b and between c and d. In this scenario we can compute $ntg(b,c) = 1.5$ and $ttc(b,c) = 6$ by symmetry. By transitivity we get the relation between b and d: $ntg(b,d) = 3$ and $ttc(b,d) = 8$. Also by symmetry we see that $ntg(d,c) = -2.4$ and by transitivity $ntg(d,b) = -4.8$ and $ttc(d,b) = 8$. What does this tell us about the driving situation? By comparing these values to Figures 1 and 2 we see that b and c are driving in the same direction and that b is approaching c. Similarly, c and d drive in the same direction with c approaching d from behind. Thirdly, the same holds for b and d. And we can even compute the speed ratios of all pairs of cars by Lemma 1, e.g., the speed ratio of b and c is $1 - \frac{-2}{6} = 1\frac{1}{3}$ and the ratio of b and d is $1 - \frac{-4.8}{8} = 1\frac{3}{5}$, i.e., b drives 33.3 % faster than c and 60 % faster than d.

Time

Now we investigate how NTG and TTC change over time. After a time period t, TTC obviously reduces by t. The NTG change depends on t and the speed ratio of b and c.

For this subsection we introduce two macros for the NTG and TTC after a time period t:

$$tntg(b,c,t) \overset{\text{def}}{=} \tfrac{x(c) + t \cdot v(c) - (x(b) + t \cdot v(b))}{v(b)}$$
$$tttc(b,c,t) \overset{\text{def}}{=} \tfrac{x(c) + t \cdot v(c) - (x(b) + t \cdot v(b))}{v(b) - v(c)}$$

By simple arithmetic transformations and Lemma 1 we get the following two results:

Proposition 7 (Temporal Evolution of NTG)

$$\mathcal{E} \models \eta(b,c) \supset tntg(b,c,t) = ntg(b,c) - t \cdot \tfrac{ntg(b,c)}{ttc(b,c)}$$

Proposition 8 (Temporal Evolution of TTC)

$$\mathcal{E} \models \kappa(b,c) \supset tttc(b,c,t) = ttc(b,c) - t$$

Acceleration

We now show how acceleration affects NTG and TTC. By acceleration we mean a discontinuous change of a car's velocity, i.e., the effect of acceleration occurs *instantaneously*. We chose this simplification for complexity reasons as we would otherwise have to deal with quadratic equations.

For this subsection we introduce macros for the NTG and TTC after an acceleration by q of b (subscript 1) and c (subscript 2):

$$\mathit{antg}_1(b,c,q) \overset{\text{def}}{=} \tfrac{x(c)-x(b)}{q\cdot v(b)} \qquad \mathit{attc}_1(b,c,q) \overset{\text{def}}{=} \tfrac{x(c)-x(b)}{q\cdot v(b)-v(c)}$$
$$\mathit{antg}_2(b,c,q) \overset{\text{def}}{=} \tfrac{x(c)-x(b)}{v(b)} \qquad \mathit{attc}_2(b,c,q) \overset{\text{def}}{=} \tfrac{x(c)-x(b)}{v(b)-q\cdot v(c)}$$

The effect of acceleration on NTG is trivial:

Proposition 9 (Effect of acceleration on NTG)

$$\mathcal{E} \models \kappa(b,c) \land q \neq 0 \supset \mathit{antg}_1(b,c) = \tfrac{1}{q} \cdot \mathit{ntg}(b,c)$$
$$\mathcal{E} \models \kappa(b,c) \supset \mathit{antg}_2(b,c) = \mathit{ntg}(b,c)$$

The effect of acceleration on TTC follows directly from Lemma 5:

Proposition 10 (Effect of acceleration on TTC)

$$\begin{aligned}&\mathcal{E} \models \eta(b,c) \land q \neq 1 - \tfrac{\mathit{ntg}(b,c)}{\mathit{ttc}(b,c)} \supset \\ &\qquad \mathit{attc}_1(b,c) = \frac{1}{(q-1)\cdot\frac{\mathit{ttc}(b,c)}{\mathit{ntg}(b,c)} + 1} \cdot \mathit{ttc}(b,c) \\ &\mathcal{E} \models \eta(b,c) \land q \neq (1 - \tfrac{\mathit{ntg}(b,c)}{\mathit{ttc}(b,c)})^{-1} \supset \\ &\qquad \mathit{attc}_2(b,c) = \frac{1}{(1-q)\cdot\frac{\mathit{ttc}(b,c)}{\mathit{ntg}(b,c)} + q} \cdot \mathit{ttc}(b,c)\end{aligned}$$

The constraints on q cover division by zero.

Basic Action Theory

With these results in hand we are ready to define a concise action theory in the situation calculus (McCarthy 1963; Reiter 2001). The situation calculus is a sorted first-order language to reason about dynamic systems with actions and change. While there are other action formalisms dealing with continuous change like the event calculus (Shanahan 1990) and Sandewall's features and fluents approach (Sandewall 1989), the situation calculus has the advantage that it comes equipped with the well-established action programming language Golog (Levesque et al. 1997), which allows a very natural encoding of agent behavious such as cars passing each other. In the situation calculus a dynamic system is modeled in terms of a *basic action theory* (BAT) which formalizes the basic relationships of *primitive actions* and situation dependent predicates and functions, called *fluents*. A situation is either the initial situation S_0 or a term $do(a,s)$ where s is the preceding situation and a is an action executed in s. The main components of a BAT are (1) a description of the initial situation S_0, (2) precondition axioms $\mathit{Poss}(a,s) \equiv \rho$ that specify whether or not the primitive action a is executable in situation s, and (3) successor state axioms which define how fluents evolve in new situations. A successor state axiom (SSA) for a fluent $F(\vec{x},s)$ has the form $F(\vec{x},do(a,s)) \equiv \gamma_F^+(\vec{x},a,s) \lor F(\vec{x},s) \land \neg\gamma_F^-(\vec{x},a,s)$ where γ_F^+ and γ_F^- describe the positive and negative effects on fluent F, respectively.

We now define a BAT $\mathcal{D}$ that unifies the previous results of this section in a single framework. From now on we assume that NTG and TTC have an additional parameter s for the current situation. Our action theory consists of two primitive actions: $\mathit{accel}(b,q)$ represents an acceleration of b by factor q; $\mathit{wait}(t)$ induces a lapse of time period t. Thus a situation $do(\mathit{accel}(b,0.5), do(\mathit{wait}(3), do(\mathit{accel}(b,2),S_0)))$ represents that b discontinuously doubles its speed, drives at this speed for 3 seconds, and then again halves its speed.

The initial situation S_0 needs to axiomatize the initial traffic scenario. A typical scenario might be that for each neighboring pair of cars we have measurements of NTG and TTC. By symmetry and transitivity a full view of the situation can be computed as in Figure 3.

The preconditions for accel and wait are simply

$$\begin{aligned}&\mathit{Poss}(\mathit{accel}(b,q),s) \equiv \text{True}, \\ &\mathit{Poss}(\mathit{wait}(t),s) \equiv t \geq 0\end{aligned}$$

The SSA for NTG can use Propositions 7 and 9 to formalize the effect of time and acceleration:

$$\begin{aligned}&\mathit{NTG}(b,c,r,do(a,s)) \equiv \\ &\quad (\exists t)\,.\,a = \mathit{wait}(t) \land \mathit{ttc}(b,c,s) \neq 0 \land \\ &\qquad r = \mathit{ntg}(b,c,s) - t \cdot \tfrac{\mathit{ntg}(b,c,s)}{\mathit{ttc}(b,c,s)} \;\lor \\ &\quad (\exists q)\,.\,a = \mathit{accel}(b,q) \land q \neq 0 \land \\ &\qquad r = \tfrac{1}{q} \cdot \mathit{ntg}(b,c,s) \;\lor \\ &\quad \mathit{NTG}(b,c,r,s) \land (\forall t)a \neq \mathit{wait}(t) \land \\ &\quad (\forall q)a \neq \mathit{accel}(b,q)\end{aligned}$$

The SSA for TTC similarly uses Propositions 8 and 10:

$$\begin{aligned}&\mathit{TTC}(b,c,r,do(a,s)) \equiv \\ &\quad (\exists t)\,.\,a = \mathit{wait}(t) \land r = \mathit{ttc}(b,c,s) - t \;\lor \\ &\quad (\exists q)\,.\,a = \mathit{accel}(b,q) \land \mathit{ntg}(b,c,s) \neq 0 \land \\ &\qquad q \neq 1 - \tfrac{\mathit{ntg}(b,c,s)}{\mathit{ttc}(b,c,s)} \land \\ &\qquad r = \frac{1}{(q-1)\cdot\frac{\mathit{ttc}(b,c,s)}{\mathit{ntg}(b,c,s)} + 1} \cdot \mathit{ttc}(b,c,s) \;\lor \\ &\quad (\exists q)\,.\,a = \mathit{accel}(c,q) \land \mathit{ntg}(b,c,s) \neq 0 \land \\ &\qquad q \neq \left(1 - \tfrac{\mathit{ntg}(b,c,s)}{\mathit{ttc}(b,c,s)}\right)^{-1} \land \\ &\qquad r = \frac{1}{(1-q)\cdot\frac{\mathit{ttc}(b,c,s)}{\mathit{ntg}(b,c,s)} + q} \cdot \mathit{ttc}(b,c,s) \;\lor \\ &\quad \mathit{TTC}(b,c,r,s) \land (\forall t)a \neq \mathit{wait}(t) \land \\ &\qquad (\forall q)\,.\,a \neq \mathit{accel}(b,q) \land a \neq \mathit{accel}(c,q)\end{aligned}$$

Recall that the notation $\mathit{ntg}(b,c,s)$ and $\mathit{ttc}(b,c,s)$ is just a shorthand for expressions involving $\mathit{NTG}(b,c,r,s)$ and $\mathit{TTC}(b,c,r,s)$, respectively, as introduced at the beginning of the section.

Note that when $\mathit{ntg}(b,c,s)$ and/or $\mathit{ttc}(b,c,s)$ are undefined so are $\mathit{ntg}(b,c,do(a,s))$ and $\mathit{ttc}(b,c,do(a,s))$. Sensing (Bacchus, Halpern, and Levesque 1999) could be used to gain new NTG and TTC information in this case.

Additionally, NTG and TTC can become undefined due to division by zero, e.g., $\mathit{ttc}(b,c,do(\mathit{accel}(b,q),s))$ is undefined if $q = 1 - \frac{\mathit{ntg}(b,c,s)}{\mathit{ttc}(b,c,s)}$ or $\mathit{ntg}(b,c,s) = 0$. Why is that? In the first case b accelerates to the same speed as c (Lemma 1) and will therefore never catch up with c. In the second case $ntg(b, c, s) = 0$ indicates that b and c are driving next to each other (but not necessarily at the same speed). Intuitively one might argue that $ttc(b, c, do(accel(b, q), s)) = 0$ should still hold, because no time passed since s and thus the TTC must remain the same. However, the further development of TTC is not clear: b might have changed its velocity to c's (which we cannot detect because $1 - \frac{ntg(b,c,s)}{ttc(b,c,s)}$ involved a division by zero), which means that b will remain next to c; thus even future $wait(t)$ actions should leave the TTC of b and c unchanged at zero in this case!

Instead of considering all these special cases in the SSAs, we can exploit transitivity at this point to avoid NTG and TTC unnecessarily becoming undefined. For space reasons we only sketch the idea. In the SSA for NTG we can extend the treatment of $a = wait(t)$ by a case that covers $ttc(b, c, s) = 0$. In that case we look for a third car d for which we can compute the NTG and TTC of the pairs b, d and d, c, respectively. Thus $ntg(b, c, do(wait(t), s))$ can be determined by transitivity via d. Analogously the SSA for TTC can be extended to cover the cases that $ntg(b, c, s) = 0$ for $a = accel(b, q)$ and $a = accel(c, q)$ by computing the NTG and TTC with a third car d and then transitively combining these values.

To conclude we elaborate on how the initial situation S_0 should look like. While the SSAs of NTG and TTC directly use Propositions 7, 8, 9, 10 to account for acceleration and time, they do not explicitly enforce symmetry and transitivity. These are in fact *state constraints* which should be satisfied in any reachable situation. This can be shown by induction on situations (Reiter 2001) to give the following result:

Proposition 11 Let $\phi(s)$ be the conjunction of the symmetry and transitivity Propositions 2, 3, 4, 6 with added situation arguments. Furthermore let $\psi(s)$ assert that NTG and TTC represent partial functions. If these properties hold in S_0, they are preserved by the successor state axioms:

$$\mathcal{D} \models \phi(S_0) \land \psi(S_0) \land executable(s) \supset \phi(s) \land \psi(s)$$

where $executable(s)$ asserts that all action preconditions in s are satisfied.

Typically, initially a chain of NTG and TTC measurements for neighboring cars is known. An appropriate initial situation can be specified by listing these measurements and adding $\phi(S_0)$ and $\psi(S_0)$ for the symmetric and transitive closure. The situation depicted in Figure 3 could thus be specified by

$$\begin{aligned}\mathcal{D}_{S_0} = \{&ntg(C, B, S_0) = -2, ttc(C, B, S_0) = 6,\\ &ntg(C, D, S_0) = 2, ttc(C, D, S_0) = 12,\\ &\phi(S_0), \psi(S_0)\}\end{aligned}$$

Of course this requires the measurements to be consistent with the symmetry and transitivity laws, otherwise the initial situation is inconsistent. E.g., if we had another measurement $ntg(B, C) = 1$ in Figure 3 this would be inconsistent with $ntg(B, C) = 1.5$ computed by symmetry.

Plan Recognition

In this section we sketch how this model can be employed in a plan recognition setting. Our approach (Schwering et al. 2012) to plan recognition is as follows. First the domain is modeled in terms of a situation calculus basic action theory which basically defines the available primitive actions and fluents. Then one defines a plan library which consists of Golog (Levesque et al. 1997) programs. To perform a concrete plan recognition task, execution of these programs is simulated. The effects in this simulation are compared to the observations of the real world. If the observations and the simulation are consistent, the respective program is considered a potential explanation of the agent's action, i.e., a recognized plan. Through concurrency we can account for multiple agents acting in parallel and interacting with each other in our simulation as in reality.

Executing a Golog program essentially means to build up a situation term. The program may involve certain nondeterministic constructs which allow the interpreter to make a choice on its own. Generally the interpreter will try to avoid actions whose precondition is not satisfied in the current situation at which point the execution halts. In the plan recognition application one seeks to explain as many observations as possible, so the interpreter looks for an execution that does best with respect to this criterion. For details on the semantics of our Golog dialect we refer to (Schwering et al. 2012). At this point we just mention the following programming constructs: Primitive actions a lead from situation s to $do(a, s)$ if $Poss(a, s)$ holds. Test actions $\phi?$ are executable only if ϕ holds in the current situation. $\delta_1; \delta_2$ represents sequential execution, $\delta_1 \parallel \delta_2$ stands for concurrency by interleaving. Further nondeterministic constructs include choice of argument $(\pi v)\ \delta(v)$ and iteration δ^*.

For the plan recognition task we have extended the BAT $\mathcal{D}$ presented in the previous section by a simple notion of lanes. Lanes are numbered consecutively, and each car b drives on exactly one of the lanes, $lane(b, s) = \ell$. A car b can instantaneously change lanes by executing an action $lc(b, \ell')$ where ℓ' denotes a lane adjacent to ℓ.[2] The longitudinal model consisting of NTG and TTC is independent from this lateral model.

To deal with NTG and TTC in a more intuitive way we have defined a few typical ranges. One such category for NTG is $close\_behind$ which stands for the NTG interval $[1, 2.5]$. Thus b is considered close behind c if $1 \leq ntg(b, c) \leq 2.5$ holds. Farther distances correspond to bigger intervals. Similarly we defined categories for TTC such as $contracting\_fast$ whose TTC interval is $[0, 5]$. We consider an observation matched by our model when there is a common category for all NTG and TTC values and all cars are in the same lane in the simulation as in reality.

With this model it is easy to express a number of driving maneuvers. The following $follow(b, c)$ program expresses that b is closely behind c and drives at the same speed as

[2] In (Schwering et al. 2012) we presented a more sophisticated model of lateral movement which measures the oscillations of a driver. We omit this here to keep the presentation simple.

c (it accelerates by $v(c)/v(b)$, c.f. Lemma 1):

$$\begin{aligned} \mathit{follow}(b,c) \overset{\text{def}}{=} & (\mathit{lane}(b) = \mathit{lane}(c))?; \\ & \mathit{accel}(b, 1 - \tfrac{\mathit{ntg}(b,c)}{\mathit{ttc}(b,c)}); \\ & (\mathit{ntg}(b,c) \in \mathit{close\_behind})? \end{aligned}$$

Another typical example is the passing maneuver:

$$\begin{aligned} & \mathit{overtake}(b,c) \overset{\text{def}}{=} \\ & \quad (\mathit{lane}(b) = \mathit{lane}(c))?; \\ & \quad (\mathit{ntg}(b,c) \in \mathit{behind})?; \\ & \quad (\quad \mathit{lc}(b, \mathit{lane}(b) + 1); \\ & \quad \quad \mathit{wait}(\mathit{until}(\mathit{ntg}(b,c) \in \mathit{in\_front})); \\ & \quad \quad \mathit{lc}(b, \mathit{lane}(b) - 1) \\ & \quad \| \quad ((\pi q)\ ((q \geq 1)?; \mathit{accel}(b,q)))^*); \\ & \quad (\mathit{ntg}(b,c) \in \mathit{in\_front})? \end{aligned}$$

Besides the initial and final tests this program consists of two concurrent threads: the first takes care of the lane changing and the actual passing, the second takes care of acceleration. The first thread asserts that at some time between the lane changes b needs to actually have passed c. This point in time is determined by the expression $\mathit{until}(\mathit{ntg}(b,c) \in \mathit{in\_front})$. We do this by a simple search: as the effect of time on NTG is linear, we can interpolate to find some representative from $\mathit{in\_front}$. TTC is linear in time, too, so we can proceed in the same way for TTC-based waiting conditions. The second thread allows to approximate the real car's acceleration with a sequence of accelerations.

Similar programs can be written to express that one car approaches the other or goes past a vehicle without any lane change. Another program is $\mathit{imitate}(b,c)$ which makes b to copycat all actions of c. This could be useful to model convoys.

Evaluation

We have implemented the described model as part of a plan recognition system in Haskell.[3] We evaluated the system with a driving simulation, TORCS,[4] to recognize driving maneuvers. Twice a second the driving simulation measures the NTG and TTC of the cars on the road and transmits this observation to the plan recognition system, which in turn tries to match these observations with a program from the plan library.

In our experiments the system reliably correctly classified follow maneuvers, passing maneuvers without lane changes, and passing maneuvers with lane changes, each with two cars on the road. The various maneuvers took about 10 s to 20 s. The recognition was done online in real time.

A more complex test run involved three cars as depicted in Figure 4: one is driving in the right lane rather slowly (d), a second one is going to pass it on the left lane (c), and a third car (b) comes from behind at excessive speed. This scenario is interesting from the driving safety point of view because there are multiple possible outcomes: To avoid a crash, b may brake strongly, swing out behind c and slowly pass d. Alternatively it may try to pierce through the gap between c and d, thus first passing c on the right lane and then pass d. These scenarios can be modeled using variants of the $\mathit{overtake}$ program described above. In our tests the driving scenario took about 20 s and the classification ran in real time.

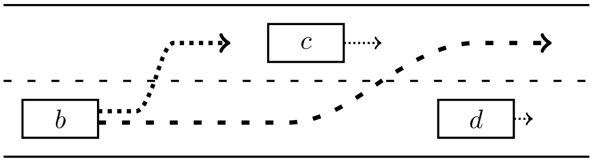


Figure 4: While c passes d, b may choose between two maneuvers to avoid a crash.

In our experiments the following issue required special treatment. When in our model two cars drive at the same speed like in the *follow* program, the TTC is obviously undefined due to division by zero. In the real world, however, cars usually do not drive at the very same speed. Instead they permanently balance their speeds so that they have the same average speed over time. This translates to large TTCs. To deal with this issue we introduced a category stable, which applies when either the TTC is undefined or the relative velocity of both cars differs by less than 2 %.

Compared to the naive global model we used in (Schwering et al. 2012), the model presented in this paper is much more tolerant with regard to velocities. E.g., the *follow* program asserts that b is closely behind c and lets b drive at the same speed as c. When both b and c accelerate a bit in the real world, the model is still consistent with the real world. In our old model, however, both cars were represented as points with absolute coordinates and absolute velocity. The real world and the model were matched just by comparing the absolute positions of the cars in reality and the model. When both cars accelerate by, say, 1 m/s in this scenario, after 10 s the positions in the model and the real world differ already by 10 m! For this reason we essentially had to fix the velocities in all our experiments in (Schwering et al. 2012). With the new model, we had no trouble to express and handle varying speeds.

Conclusion

In this paper we proposed a new model to represent and reason about longitudinal distances between cars. By measuring relative distances in time instead of spatially this model takes after the human cognition in traffic. These temporal measures are more independent of the pace than spatial measures. Our model exclusively relies on relative data and no global information at all, which makes it realistic for real world applications. From this relative information it still provides a fairly complete view of the traffic situation. We also integrated the concepts of continuous time and acceleration in order to account for the high dynamics of the fast-

[3] http://www.haskell.org/

[4] http://torcs.sourceforge.net/

paced traffic domain.

We plan to experiment with this model in larger-scale traffic simulations to exploit the transitivity more explicitly. We also aim to integrate sensing and knowledge (Bacchus, Halpern, and Levesque 1999) into the model and plan recognition system to deal with incomplete knowledge and newly appearing cars. Another open problem is how to deal with measuring inaccuracies of real applications and the resulting inconsistencies with symmetry and transitivity.

Acknowledgments

We thank the anonymous reviewers for their very helpful suggestions and Stefan Schiffer for commenting on a draft of this paper. The first author is supported by the B-IT Graduate School.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26:832–843.

Bacchus, F.; Halpern, J. Y.; and Levesque, H. J. 1999. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence* 111(1-2):171–208.

Clementini, E.; Di Felice, P.; and Hernández, D. 1997. Qualitative representation of positional information. *Artificial Intelligence* 95(2):317–356.

Dagli, I.; Brost, M.; and Breuel, G. 2002. Action recognition and prediction for driver assistance systems using dynamic belief networks. In *Proceedings of the Conference on Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, 179–194. Springer-Verlag.

Helbing, D., and Nagel, K. 2004. The physics of traffic and regional development. *Contemporary Physics* 45(5):405–426.

Levesque, H.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. 1997. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming* 31:59–84.

Marsden, G.; McDonald, M.; and Brackstone, M. 2001. Towards an understanding of adaptive cruise control. *Transportation Research Part C: Emerging Technologies* 9(1):33–51.

McCarthy, J. 1963. Situations, Actions, and Causal Laws. Technical Report AI Memo 2 AIM-2, AI Lab, Stanford University, California, USA. Published in Semantic Information Processing, ed. Marvin Minsky. Cambridge, MA: The MIT Press, 1968.

Mossakowski, T., and Moratz, R. 2012. Qualitative reasoning about relative direction of oriented points. *Artificial Intelligence* 180-181:34–45.

Randell, D.; Cui, Z.; and Cohn, A. 1992. A spatial logic based on regions and connection. In Nebel, B.; Rich, C.; and Swartout, W., eds., *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*. San Mateo, California: Morgan Kaufmann. 165–176.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press.

Sandewall, E. 1989. Combining logic and differential equations for describing real-world systems. In *Proceedings of the first international conference on Principles of knowledge representation and reasoning*, 412–420. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Schwering, C.; Beck, D.; Schiffer, S.; and Lakemeyer, G. 2012. Plan recognition by program execution in continuous temporal domains. In Burgard, W.; Konolige, K.; Pagnucco, M.; and Vassos, S., eds., *Proceedings of the 8th International Cognitive Robotics Workshop*, 77–84.

Shanahan, M. 1990. Representing continuous change in the event calculus. In *Proceedings of the 9th European Conference on Artificial Intelligence*, 598–603.

Van de Weghe, N.; Cohn, A. G.; Maeyer, P. D.; and Witlox, F. 2005. Representing moving objects in computer-based expert systems: the overtake event example. *Expert Systems with Applications* 29:977–983.

Van de Weghe, N.; Cohn, A.; De Tre, G.; and De Maeyer, P. 2006. A qualitative trajectory calculus as a basis for representing moving objects in geographical information systems. *Control and Cybernetics* 35(1):97–119.

Vilain, M.; Kautz, H.; and Beek, P. 1986. Constraint propagation algorithms for temporal reasoning. In *Readings in Qualitative Reasoning about Physical Systems*, 377–382. Morgan Kaufmann.