# Time-to-contact control: improving safety and reliability of autonomous vehicles

Liang Wang

Computer Science and Artificial Intelligence Laboratory (CSAIL),
Massachusetts Institute of Technology (MIT),
Cambridge, MA 02139, USA
Email: wangliang.bjtu@gmail.com

Berthold K.P. Horn*

Department of Electrical Engineering and Computer Science (EECS),
Massachusetts Institute of Technology (MIT),
Cambridge, MA 02139, USA
Email: bkph@csail.mit.edu
*Corresponding author

**Abstract:** Under traditional car-following control, i.e., human drivers' behaviour, the stability condition of traffic system is *not* satisfied in general. For safety and reliability of autonomous vehicles, additional danger warning system must be used in the adaptive cruise control system to prevent inevitable potential collisions. One reasonable quantity of evaluating potential collisions is time to contact (TTC): how soon will potential collision occur? In this paper, we provide TTC feedback control to improve safety and reliability of autonomous vehicles, and show the effectiveness of TTC feedback. TTC can be estimated by machine vision techniques with single uncelebrated camera (i.e., passive sensors). We provide detailed mathematical analysis and algorithmic implementation. The machine vision-based TTC algorithm is pretty fast such that the whole system can be implemented on Android smart phones running in real-time. Moreover, it is not trial to estimate relative velocity by differentiating the measured distance between cars with respect to time, because inevitable measurement noise in the distance measurements will be amplified by the derivative operation. The time-to-contact-based algorithm provides an alternative approach to estimating the relative velocity, which can also be fused with measurements from other active sensors, if desired.

**Keywords:** autonomous vehicles; time to contact; TTC; danger detection; machine vision; android smartphones.

**Reference** to this paper should be made as follows: Wang, L. and Horn, B.K.P. (2020) 'Time-to-contact control: improving safety and reliability of autonomous vehicles', *Int. J. Bio-Inspired Computation*, Vol. 16, No. 2, pp.68–78.

**Biographical notes:** Liang Wang received his PhD in Computer Application Technology from the School of Computer and Information Technology, Beijing Jiaotong University. He was involved in the Mathematics Department, Massachusetts Institute of Technology (MIT), from 2011 to 2013. He was a Post-Doctoral Research Scholar with the Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT from January 2015 to December 2018. His research interests include: machine vision, inverse problems, and intelligent vehicles.

Berthold K.P. Horn is a Professor of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology (MIT). He received his BSc Eng degree from the University of the Witwatersrand in 1965 and SM and PhD degrees from MIT in 1968 and 1970, respectively. He is the author, coauthor or editor of books on the programming language *LISP* and machine vision, including *Robot Vision*. He was awarded the Rank Prize for pioneering work leading to practical vision systems in 1989 and was elected a Fellow of the American Association of Artificial Intelligence in 1990 for significant contributions to artificial intelligence. He was elected to the National Academy of Engineering in 2002 and received the Azriel Rosenfeld Lifetime Achievement Award from the IEEE Computer Society for pioneering work in early vision in 2009. His current research interests include machine vision, computational imaging and intelligent vehicles.

This paper is a revised and expanded version of a paper entitled 'Time-to-contact control for safety and reliability of self-driving cars' presented at the 2017 International Smart Cities Conference (ISC2), Wuxi, China, 14–17 September 2017.

## 1 Introduction

Driver assistance systems and autonomous vehicles come closer and closer to reality with the rapid development of sensors and wireless communications. Correspondingly, many interesting questions arise. For instance, how to control a car based on the information from the sensors? How much autonomous vehicles can improve the traffic situation, e.g., reduce the number of collisions, stabilise the traffic flow, and increase the throughput of a highway? Note that autonomous driving systems can take into account more information about the environment of the car than a human driver can, and thus show promise of yielding attractive improvements relative to today's traffic situation. See Horn (2013), Baran and Horn (2013), Wang et al. (2017) and Horn and Wang (2018) for more details. However, at least the early version of autonomous cars are still supposed and designed to drive like human drivers by many people. That is, try to following the car ahead based on local measurements, e.g., space and relative speed.

Even under the same control mode, there is still significant difference between autonomous vehicles and human-driving cars. For instance, the response of sensors is much faster than human drivers. Correspondingly, the required safe space (which is proportional to reaction time) between self-driving cars is much smaller than the one for human-driving cars. The car density in autonomous-vehicle traffic can be much higher than nowadays traffic. As human drivers, we are too familiar with the 'stop-and-go' traffic jams. Will such autonomous-vehicle traffic (with smaller safe space used for each car) be better, or even worse? First, by input/output stability analysis, we find the answer to this question:

- Under car-following control, the stability condition becomes harder and harder to satisfy as the reaction time decreasing. Thus, the traffic flow will become more unstable due to the 'tailgating ability' of autonomous vehicles.

This conclusion provides tow suggestions of designing adaptive cruise control (ACC) system:

1 stop tailgating, i.e., keeping much larger reaction time than the response time of the sensors

2 design danger preventing system to avoid the inevitable potential collisions caused by traffic system's instability.

Intuitively, the danger can be predicted depends on the quantity that how soon the potential collision will occur. This predicted collision time is called *time to contact* (TTC). Then, we provide a TTC control strategy which evaluates the potential danger of collision by the inverse of time-to-contact (1/TTC). Emergency brake will be taken immediately if 1/TTC is larger than a preset threshold. Numerical simulation shows the effectiveness of TTC feedback to prevent inevitable potential collisions (see Figure 3).

It is not trivial to estimate TTC, i.e., the ratio of space and relative speed, by active sensors. Although space can be measured by Radar or Lidar effectively, relative speed can *not* be estimated directly by taking the time derivative of the measured space. Any small noise in the distance measurement will be amplified by taking the time derivative. Machine vision provides an effective solution to this problem. As is well known, at least two calibrated cameras are needed to obtain both depth and motion information about objects in the scene using binocular stereo (see, e.g., Horn, 1986). Finding distance and velocity monocularly is difficult. However, the *ratio* of distance to velocity (i.e., time to contact) can be obtained relatively simply monocularly (see Horn et al., 2007). We provide mathematical analysis TTC estimation and the corresponding detailed algorithmic implementation. The estimated TTC is given by analytical close-form solution, rather than iterative approach. Thus, the algorithm is pretty fast and can even run on Android smartphones in real time. We also built a robot-based prototype system to test the TTC implementation.

The distance between the controlled car and the leading car are used in the ACC system. However, this information is *not* enough to drive a car safely and smoothly. Human drivers also use information about the relative velocity of the leading car with respect to their car. We have shown that the relative-velocity-based control term is *important* for stability (see, e.g., Horn, 2013; Wang et al., 2017). Thus, how to obtain stable and reliable measurements of relative velocity is an important issue for autonomous vehicles. As mentioned above, small noise in the distance measurement will be amplified by taking the time derivative. Thus, relative velocity estimated by taking the time derivative of the measured distance to the leading car can not be used directly by the ACC system. Some other, more reliable, approach to measuring relative velocity is needed. The relative speed of the vehicles can also be calculated directly by multiplication of the distance measurement with the inverse of the TTC (1/TTC) (note that the natural output of the TTC algorithm *is* 1/TTC). The output of this new method for estimating the relative velocity can then be fused with the estimate obtained by taking the derivative of the distance, using e.g., a Kalman filter, obtain an even more accurate and reliable estimate. Thus, time-to-contact feedback is very important for autonomous vehicles, not only for the car's *safety*, but also for the *reliability* of the ACC system.

## 2 Car-following control

Human driver's behaviour can be modelled as following the leading car (Lighthill and Whitham, 1955; Chandler et al., 1958; Herman et al., 1959). That is, adjust the car's speed according to the distance from the car ahead. Moreover, relative speed between the current and the leading car is also used by human drivers to adjust the car's speed. We first provide the simplified mathematical description of such human-driver's behaviour, which is

known as *car-following control* model (CFM). Then, by input/output analysis, we provide the *stability condition* for such CFM-based traffic. Finally, we shows why such stability condition is difficult to satisfied in real traffic and thus the 'phantom traffic jams' caused by human drivers appears frequently in nowadays traffic. The traffic flow instabilities are somehow inevitable in the CFM-based traffic. Thus, some collision preventing technique and danger-warning system is somehow necessary for the cars, especially for the autonomous vehicles (or autonomous driving system).

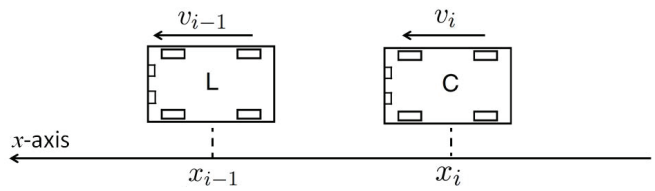
### 2.1 Mathematical model of CFM traffic

In this paper, we only consider the case of single-lane traffic. That is, a line of cars running on a road (see Figure 1). Let $x_n$ and $v_n = \dot{x}_n$ denote the position and speed of car $n$. The adjustment of the car's (say car $n$) speed, i.e., the acceleration command $a_n$, is based *only* on its forward measurements, i.e., space $d_n = (x_{n-1} - x_n - l$ (where $l$ denotes the car length) and relative speed $r_n v_{n-1} - v_n$ between the current car and the leading car. Mathematically,

$$a_n = k_d(d_n - v_n T) + k_v r_n \tag{1}$$

where, $T$ is known as the *reaction time*, $k_d \geq 0$ and $k_v \geq 0$ are known as the *proportional gain* and *derivative gain*, respectively (Horn, 2013). That is, the current car is controlled to following the car ahead with enough *safe distance* $s_n = v_n T$. Thus, equation (1) is called car-following control model (CFM). Note that CFM (1) is the simplified lumbarisation of human driver's behaviour. In real traffic, speed limitation, acceleration/deceleration limitation, desired speed, nonlinear reactions and etc also appears in the drivers' behaviour. However, the simplified linear model (1) provides us a good explanation why almost all the drivers go through the 'stop-and-go' pattern, which is also known as the 'phantom traffic jams', again and again in highway traffic.

**Figure 1** Illustration of the car-following control




Notes: The blocks with 'L' and 'C' denote the leading car and current controlled car. *Only* local measurements are used in car-following control.

### 2.2 Stability analysis of CFM traffic

When you are driving on the highway and going through a traffic jam, you might think there must be some accident far away from you. However, most of the time, there is actually *no* accident. The reason the traffic flow stops for hours, and consequently you car 'moving and waiting and moving and waiting again' is due to the *instability* of such CFM traffic. Suppose in the beginning, the traffic condition is perfect. All cars are equally spaced and move and move at the same speed. The basic question is what is the influence of the actions first car (say car 0), e.g., emergency brake or small perturbation in its speed, to the whole traffic. Will the perturbation caused by the first car be amplified again and again by the following cars? Or Will such perturbation be suppressed by the following cars and disappear finally? The answer to this question can be found by input/output stability analysis of the traffic system.

Note that $a_n = \ddot{x}_n$ and $v_n = \dot{x}_n$. Thus, equation (1) leads to the following ordinary differential equation (ODE)

$$\ddot{x}_n + (k_v + T)\dot{x}_n + k_d(x_n + l) = k_v \dot{x}_{n-1} + k_d x_{n-1} \tag{2}$$

Taking *Laplace transform* of both sides in equation (2), we find:

$$H(s) = \frac{X_n(s)}{X_{n-1}(s)} = \frac{k_v s + k_d}{s^2 + (k_v + T)s + k_d} \tag{3}$$

$H(s)$ is called the *transfer function*. We then find

$$X_n(s) = H(s)X_{n-1}(s) = \cdots = H^n(s)X_0(s) \tag{4}$$

$X_0(j\omega)$ is the magnitude of sinusoid component $e^{j\omega t}$ in the perturbation of the first car. The (string) stability of the system implies that

$$\lim_{n\to\infty} X_n(j\omega) = 0 \tag{5}$$

for all $\omega \neq 0$. Thus, for stability, we must have

$$\|H(j\omega)\| \leq 1 \tag{6}$$

for sinusoidal excitation of *any* frequency $\omega \neq 0$.

Note that

$$\|H(j\omega)\|^2 = \frac{k_d^2 + k_v^2\omega^2}{(k_d - \omega^2)^2 + (k_v + T)^2\omega^2} \tag{7}$$

The stability condition (6) corresponds then to

$$k_d^2 + k_v^2\omega^2 \leq (k_d - \omega^2)^2 + (k_v + T)^2\omega^2 \tag{8}$$

for all $\omega^2 \neq 0$. That is,

$$k_d T^2 + 2k_v T \geq 2 \tag{9}$$

Equation (9) is difficult to be satisfied in general. Thus, we are too familiar with the 'phantom traffic jams' in nowadays highway traffic. In summary, the 'stop-and-go' pattern is actually generated by 'tailgating' behaviours, i.e., using small $T$, of human drivers.

### 2.3 Control system design

For human drivers, the 'reaction time' $T$ is usually taken to be about 1 second (Horn, 2013). For autonomous vehicles, the reaction time $T$ could be much smaller, e.g., 0.1 sec. Thus, we can predict the traffic under autonomous cars will be come even worse if CFM is used to build the adaptive cruise control (ACC) system of the autonomous

vehicles, because the autonomous vehicles have the ability to tailgate even more. The above stability analysis provides two suggestions of designing ACC system for autonomous vehicles:

1 *Stop tailgating*: choosing larger $T$, e.g., greater than 1 sec. even if the sensors' reaction time is much smaller than that.

2 Collision alert system is *necessary*: system's instability implies that the potential collision between successive cars is somehow *inevitable* for the simplified linear control strategy (1). Thus, some additional collision alert and preventing technique *must* be used for the safety and reliability of autonomous vehicles.

In real application, the reaction time $T$ can not be set as an arbitrary large number. If $T$ is too larger, other cars from neighbouring lanes may merge in. Some new control strategy tailoring autonomous vehicles may relax system's instabilities. However, at least in the beginning of the age of autonomous vehicles, we can imagine that most autonomous vehicles might be built to drive like human drivers. Thus, the above two suggestions are important to take into account for ACC system design at least in the early version of future autonomous vehicles.

## 3 Time-to-contact control

By stability analysis, we predict the inevitable of collision between successive vehicles under simplified linear control strategy (1), and show the necessary of additional collision alert and preventing system for ACC design. The next question is how to design such alert system to prevent car-collisions. Note that relative position, i.e., space $d_n$ and speed difference $r_n$ are used by human drivers in CFM (1). If the space $d_n$ is small, there might be potential danger of collision. Also, if the relative velocity $-r_n$ is large, there might be potential danger of collision. However, these two conditions are *not* sufficient to indicate potential danger. The potential danger should be evaluated by *both* space $d_n$ and relative velocity $-r_n$ between the successive two vehicles. A more reasonable quantity to evaluate the potential danger is their ratio, i.e.,

$$TTC = \frac{d_n}{-r_n} = \frac{x_{n-1} - x_n - l}{v_n - v_{n-1}} \tag{10}$$

That is, 'how soon will the potential collision happen?' This quantity is called *time to contact* (TTC). Intuitively, if TTC is a small positive number, then the car should break hardly to prevent the potential collision.

### 3.1 TTC-based emergency brake

First, we can try to add an additional TTC-based feedback control component to the ACC system. Emergency brake should be taken when TTC is small (and positive) rather than when TTC is larger. Thus, the inverse of time-to-contact (1/TTC) should be used as a feedback to the control system. Moreover, a threshold of 1/TTC should also be used. That is

$$\lfloor 1/TTC \rfloor_+ = \begin{cases} 1/TTC, & \text{if } 1/TTC > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Now, the control strategy becomes:

$$a_n = \begin{cases} k_d(d_n - v_n T) + k_v r_n, & \text{if } \lfloor 1/TTC \rfloor_+ < \eta \\ a_{\min}, & \text{otherwise.} \end{cases} \tag{12}$$

where $\eta$ is a positive threshold and $a_{\min}$ is the largest deceleration for the car. (In general $a_{\min} = -5$ m/s$^2$.)

### 3.2 Simulation

Now, the control system (12) is *nonlinear*. The input/output analysis for the linear CFM (1) can not be used directly here. We build a Java-based simulation platform to demonstrate the effectiveness of TTC emergency brake. Figure 2 shows the interface of the simulator. Suppose there are totally $n$ cars running on a circle (with total length as $L = 500$ metres). Users can easily specify the environment of the simulation, e.g., the car number $n$, the desired speed, speed limitation, acceleration/deceleration limitation, and the parameters used in ACC system, e.g., feedback gains $k_d$ and $k_v$, reaction time $T$, threshold $\eta$ in (12) for TTC control.

**Figure 2** The interface of the simulation platform (see online version for colours)


Note: Users can specify the environment of the simulation and set the parameters used in ACC system easily.

The numerical approach used for simulation is as following:

- The time-step is set as $\Delta t = 0.1$ sec. In each iteration, first, calculating the acceleration $a_n$ by

equation (12). Then calculate the new position $x_n^{\text{new}}$ and speed $v_n^{\text{new}}$ of car $n$ by

$$x_n^{\text{new}} = x_n^{\text{old}} + v_n^{\text{old}}\Delta t + \frac{1}{2}a_n(\Delta t)^2 \tag{13}$$

$$v_n^{\text{new}} = v_n^{\text{old}} + a_n\Delta t \tag{14}$$

Then, calculating the new acceleration $a_n$ by equation (12) again, and then renew the position and speed of car $n$.

- If car collision happen, i.e., $x_{n-1} - x_n < l$, then let the following car stop suddenly and start moving again, i.e., setting $x_n = x_{n-1} - l$ and $v_n = 0$.
- Circular modulo: if $x_n > L$ (with $L = 500$ m), then set $x_n = x_n - L$. (Note that $d_{n+1}$ should be adjusted by adding $L$.)

Figure 3 shows some simulation results. The curves in Figure 3 correspond to the 'trajectories' of the (totally 22) cars in the space-time domain. The horizontal axis is the space (with total range as 500 m), and the vertical axis is the time (with total range as 25 sec.) . The speed of the car is the inverse of the slop of the corresponding curve. When car collisions occur, the slop of the corresponding cure changes suddenly. The parameters are the ones in Figure 2, except the 'rhreshold: 1/TTC', i.e., $\eta$, is adjusted to show the effect of TTC emergency brake.

In the beginning, the traffic is almost perfect. Very small perturbations are introduced in the cars' state [see Figure 3(a)]. Traffic flow under CFM is unstable. Thus, the tinny perturbation is amplified to cause traffic jams. Without TTC control, i.e., setting large 'threshold: 1/TTC' as $\eta = 10$, car collision is inventible due to the instability of the traffic system [see Figure 4(b)] We then set 'threshold: 1/TTC' as $\eta = 0.5$, i.e., breaking hard 2 sec. before potential collision, the TTC emergency brake control prevents car collisions effectively [see Figure 3(c)].

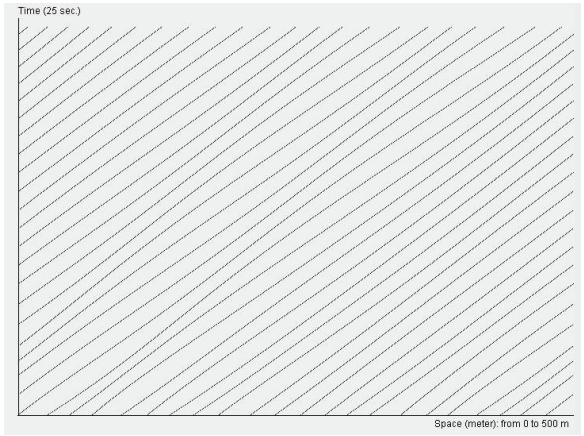
Note that the traffic flow is still *not* smooth by adding TTC emergency brake, because the traffic system is unstable. TTC feed back can prevent potential car collisions and improve the vehicles' safety and reliability. However, 'phantom traffic jams' are not suppressed effectively by TTC emergency brake, and the cars still move in the 'stop-and-go' pattern. Actually, Figure 3(c) is closer to our experiences of highway traffic. That is, follow the leading car and break hard when potential danger might occur. Consequently, drive in the periodic pattern of 'speed up, slow down and speed up again'.

### 3.3 TTC feedback control

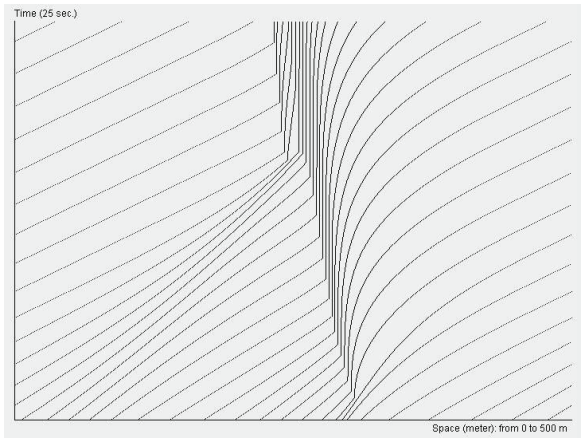
Human drivers are actually estimating TTC (or 1/TTC) all the time during their driving. Another control strategy is to adjust the vehicle's state using TTC estimation through feedback control. For instance, another feedback gain $k_{\text{TTC}} > 0$ is used also to determine the acceleration/deceleration, i.e.,

$$a_n = k_d(d_n - v_nT) + k_v r_n - k_{\text{TTC}}\lfloor 1/TTC \rfloor_+ \tag{15}$$

**Figure 3** The simulation results, (a) in the beginning, the traffic is almost perfect (b) traffic flow instability will cause car collision (c) TTC emergency brake prevent car collisions effectively




(a)



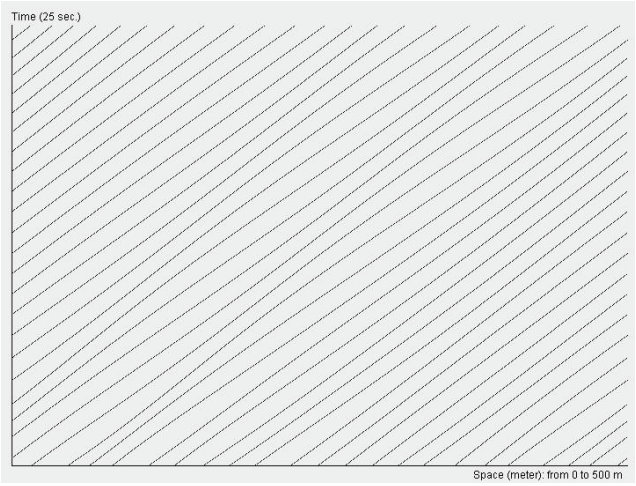

(b)


(c)

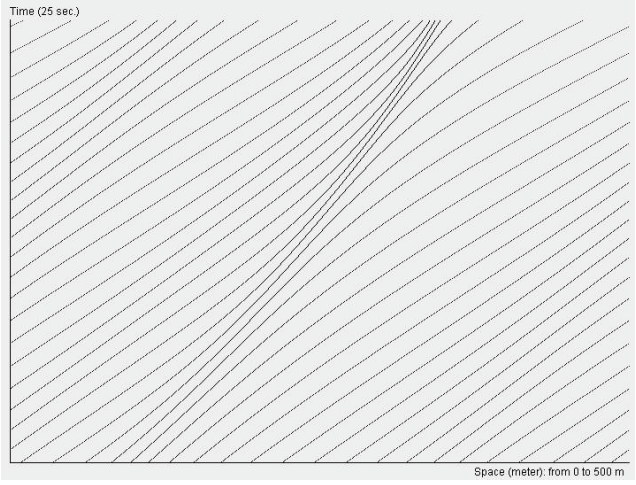
Notes: The curves correspond to the 'trajectories' of the (totally 22) cars in the space-time domain.
a In the beginning, very small perturbations are introduced in the cars' state.
b Without TTC control, car collision is inventible due to the instability of the traffic system.
c By adding TTC emergency brake, i.e., setting small 'Threshold: 1/TTC' as $\eta = 0.5$, potential car collisions are prevented effectively.

Different from the simplified car-following control in (1), equation (15) is nonlinear. Thus, we can not provide the above Laplace-transform-based analysis. Here, we used algorithm in Section 3.2 to simulate the corresponding traffic flow. Figure 4 shows some simulation results.

**Figure 4** The simulation results, (a) in the beginning, the traffic is almost perfect (b) traffic flow instability will cause car collision




(a)




(b)

Notes: The curves correspond to the 'trajectories' of the (totally 22) cars in the space-time domain. TTC feedback in (15), with $k_{\mathrm{TTC}} = 0.3$, is used to prevent potential car collision.

Similar to the results in Figure 3, the traffic system is *not* stable. In the beginning, the traffic flow is almost smooth. However, small perturbations are amplified continuously to generate traffic jams. TTC feedback control (15) prevents the potential car collisions. More complicated models can also be built. For instance, both TTC feedback control and TTC emergency brake can be implemented simultaneously.

The left problem is how to estimate TTC effectively. Although the space $d_n$ can be obtained reliably using some other sensor like Radar or Lidar. However, if the relative velocity $r_n = v_{n-1} - v_n$ is estimated by taking the derivative of $d_n$ directly, i.e.,$r_n = d/dt(d_n)$, then the noise in $d_n$ will be amplified. If, for example, we model small components of perturbations in the measurements as waves of the form $\epsilon \sin(\omega t)$, then the derivative will be corrupted by $\omega\epsilon \cos(\omega t)$, and so higher frequency components of measurement noise will be amplified a lot. (We could attempt to limit this effect by approximate low pass filtering the result, but that would introduce latency or time delays, which are, of course, not good for stability of control systems.)

Note that human drivers do *not* estimate TTC by the above approach. Space and speed difference are *not* estimated separately or independently. Actually, human drivers estimate TTC somehow *directly* through vision (i.e., a passive sensor rather active sensor as Radar or Lidar). We need explore the vision-based direct approach to estimate TTC (or 1/TTC) effectively.

## 4 Vision-based time-to-contact estimation

As an object approaches you, its image on your retina will expand; conversely, as it moves further away, the image will become smaller and smaller. This observation gives us some intuition into how one might estimate the motion of an object from its time-varying image. As is well known, at least two calibrated cameras are needed to obtain the depth of objects in the scene. Thus, the absolute motion of the object, i.e., the speed, can not be estimated using just a single camera. However, the *ratio* of the object's depth and its speed, which is known as time to contact (TTC), *can* be estimated using a single camera. See Horn et al. (2007) for more details.

### *4.1 Passive navigation*

We chose to use the camera coordinate system shown in Figure 5. The origin is at the pin-hole of the camera, and the $Z$ axis is along the optical axis of the camera. The $X$ and $Y$ directions in 3-D are aligned with the $x$ and $y$ axes of the image sensor (Horn, 1986). Suppose there is a planar surface perpendicular to the $Z$ axis, and moving at the speed $(U, V, W)$. The position of the image $(x, y)$ of the point $(X, Y, Z)$ is determined by the **perspective projection** equation (see e.g., Horn, 1986), i.e.,

$$x = f\frac{X}{Z} \qquad \text{and} \qquad y = f\frac{Y}{Z} \tag{16}$$

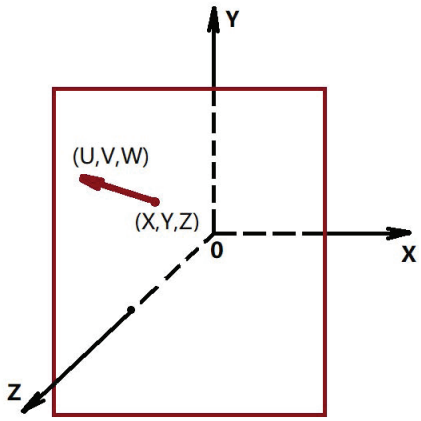
The motion $(U, V, W)$ and the depth $Z$ will generate movements of the image pattern, which is called the *optical flow* $(u, v)$ (see e.g., Horn, 1981). The relationship between motion in the world and motion of corresponding image points is given by (see Bruss and Horn, 1983):

$$u = \frac{Uf - xW}{Z} \qquad \text{and} \qquad v = \frac{Vf - yW}{Z} \tag{17}$$

The problem of estimating motion $(U, V, W)$ and depth $Z(x, y)$ from $(u, v)$ is known as **passive navigation** in the machine vision field. See e.g., Bruss and Horn (1983), Horn

and Weldon (1988) andShahriar and Horn (1987) for more details. Even for the very simple case shown in Figure 5, there is *no* unique solution. See e.g., Horn (1986) and Bruss and Horn (1983) for more detailed analysis.

**Figure 5** We use the camera coordinate system, in which the $Z$ axis is along the optical axis of the camera (see online version for colours)




Notes: A planner surface is perpendicular to the $Z$ axis, and moving at the speed $(U, V, W)$. The TTC is $Z/(-W)$.

Note that the *time to contact* (TTC) is $Z/(-W)$. Also, the *focus of expansion* (FOE), denoted by $(x_0, y_0)$ is (McQuirk et al., 1998):

$$x_0 = f\frac{U}{W} \qquad \text{and} \qquad y_0 = f\frac{V}{W} \tag{18}$$

Thus, equation (17) can be written as:

$$u = (x - x_0)C \qquad \text{and} \qquad v = (y - y_0)C \tag{19}$$

where $C = 1/\text{TTC}$. The three parameters $(x_0, y_0)$ and $C$ can be estimated from the given optical flow $(u, v)$.

### 4.2 Optical flow

The problem of estimating $(u, v)$ from the changes in the image pattern – which is described by spatial variation $(E_x, E_y)$ and temporal variation $E_t$ of an image sequence $E(x, y, t)$ — is known as the *optical flow* problem (see, e.g., Horn, 1981). Here, we face a very special case in which $(u, v)$ is determined by only 3 parameters. Thus, it is more efficient to solve this problem using the *optical flow constraint* directly, i.e.,

$$uE_x + vE_y + E_t = 0 \tag{20}$$

Substituting (19), we find

$$E_xA + E_yB + GC + E_t = 0 \tag{21}$$

where $G = xE_x + yE_y$, $A = x_0C$ and $B = y_0C$. Note that here $E_x$, $E_y$, $E_t$ and $G$ are calculated from the image sequence, while the parameters $A$, $B$, $C$ are the unknowns to be determined.

### 4.3 Least-square solution

We can solve for $A$, $B$ and $C$ in (21) by a *least squares* method. That is, we minimise the following objective function

$$\min_{A,B,C} \iint (E_xA + E_yB + GC + E_t)^2 \, dx\, dy \tag{22}$$

That coincides with solving a $3 \times 3$ linear system [see, e.g., Strang (2003)]:

$$\begin{pmatrix} a & b & c \\ b & d & e \\ c & e & g \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{23}$$

where

$$a = \iint E_x^2 dxdy \tag{24}$$

$$b = \iint E_xE_y dxdy \tag{25}$$

$$c = \iint E_xG dxdy \tag{26}$$

$$d = \iint E_y^2 dxdy \tag{27}$$

$$e = \iint E_yG dxdy \tag{28}$$

$$g = \iint G^2 dxdy \tag{29}$$

$$p = \iint E_xE_t dxdy \tag{30}$$

$$q = \iint E_yE_t dxdy \tag{31}$$

$$r = \iint GE_t dxdy \tag{32}$$

Note that only multiplication and accumulation operations are involved in generating the nine quantities that appear in equation (23), and that we can solve for $(A, B, C)$ analytically. Thus, not surprisingly, this least-square approach is much faster than e.g., some method based on feature-detection and matching. If desired, the FOE can be calculated using $(x_0, y_0) = (A/C, B/C)$.

As we conclude above that it is non-trivial to obtain a reliable measurement of relative velocity $r_n$ for use in control of autonomous vehicles. However, by such machine vision-based approach, $C = 1/TTC$ can be estimated efficiently. Thus, the relative velocity can be calculated by $r_n = d_nC$ directly. The remaining work is to implement the algorithm outlined above. Such estimated $r_n$ can be used to fuse with the relative velocity estimated from active sensors (with amplified noise) to generate a more reliable speed-difference estimation. Thus, TTC (or 1/TTC) estimation is not only important for the *safety* of the autonomous vehicles but also significant for the *reliability* of the corresponding ACC system implementation.

## 5 Implement TTC algorithm

We implement the TTC algorithm (in JAVA) on Android smartphones. We describe the implementation in detail below. Moreover, In order to stabilise the estimation of TTC and FOE, a recursive filter is also used.

### *5.1 Smart-phone system specification*

The Android TTC code contains three JAVA classes:

1 TTCCalculator: it extends the class 'Activity', and calculate TTC using image frame data

2 CameraPreview: it extends the class 'SurfaceView', and displays the previewed image from the camera

3 DrawOnTop: it extends the class 'View', and shows the intermediate result for TTC calculation.

Let $E_x(i,j;k)$, $E_y(i,j;k)$ and $E_t(i,j;k)$ be the values of $E_x$, $E_y$, $E_t$ at pixel $(i,j)$ in the image frame $k$. In the algorithm, these values are estimated (using image frame $k-1$ and frame $k$) by following:

$$\begin{aligned} E_x(i,j;k) = {} & \frac{1}{4}\left(E_x(i-1,j;k)\right. \\ & \left. - E_x(i-1,j-1;k)\right) \\ & + \frac{1}{4}\left(E_x(i,j;k) + E_x(i,j-1;k)\right) \\ & + \frac{1}{4}\left(E_x(i-1,j;k-1)\right. \\ & \left. + E_x(i-1,j-1;k-1)\right) \\ & + \frac{1}{4}\left(E_x(i,j;k-1)\right. \\ & \left. + E_x(i,j-1;k-1)\right) \end{aligned} \tag{33}$$

$$\begin{aligned} E_y(i,j;k) = {} & \frac{1}{4}\left(E_x(i,j-1;k)\right. \\ & \left. - E_x(i-1,j-1;k)\right) \\ & + \frac{1}{4}\left(E_x(i,j;k) + E_x(i-1,j;k)\right) \\ & + \frac{1}{4}\left(E_x(i,j-1;k-1)\right. \\ & \left. + E_x(i-1,j-1;k-1)\right) \\ & + \frac{1}{4}\left(E_x(i,j;k-1)\right. \\ & \left. + E_x(i-1,j;k-1)\right) \end{aligned} \tag{34}$$

$$\begin{aligned} E_t(i,j;k) = {} & \frac{1}{4}\left(E_x(i,j;k) - E_x(i,j;k-1)\right) \\ & + \frac{1}{4}\left(E_x(i-1,j;k)\right. \\ & \left. - E_x(i-1,j;k-1)\right) \\ & + \frac{1}{4}\left(E_x(i-1,j-1;k)\right. \\ & \left. - E_x(i-1,j-1;k-1)\right) \\ & + \frac{1}{4}\left(E_x(i,j-1;k)\right. \\ & \left. - E_x(i,j-1;k-1)\right) \end{aligned} \tag{35}$$

Then, $G(i,j;k)$ is calculated by:

$$G(i,j;k) = jE_x(i,j;k) + iE_y(i,j;k) \tag{36}$$

And the coefficients $a$ to $r$ in the linear equation (23) can then be calculated accordingly by replacing the double integral by double sum of the calculated values of $E_x(i,j;k)$, $E_y(i,j;k)$, $E_t(i,j;k)$ and $G(i,j;k)$. The closed form solution of the $3 \times 3$ linear equation (23) is well-known (see, e.g., Strang, 2003). Thus, the solution $A$, $B$ and $C$ can be calculated directly by (only) several multiplication and addition operations. Thus the TTC algorithm is pretty fast.

The sampling rate is set at 10 frames per second (fps). Thus the time interval is $\Delta t = 0.1$ sec. The resolution of the image frame is set to $720 \times 1080$ pixels. The image frame is down-sampled by 4, i.e., to a $180 \times 270$ matrix, as the input data $E$.

Figure 6 shows some testing results running on Huawei Nexus 6P. The first four sub-images in the top-left corner of Figure 6(b) are the down-sampled $E$, $E_t$, $E_x$, $E_y$, while the last two show the pixels' motion (in $x$ and $y$ direction, respectively) due to the phone's angular velocity — which is measured by the phone's gyroscope. The three 'bars' in Figure 6 show the computed parameters $A$, $B$, $C$, which indicate the motion (of the camera) in the $X$, $Y$ and $Z$ directions. Red means positive, while green means negative.

### *5.2 Two-registers recursive filtering*

Note that the speed of both the current controlled car and the leading car are continuous (velocity cannot change instantaneously; in fact, its rate of change is limited by the maximum accelerations and decelerations possible for the vehicle). Thus, we expect the TTC ($1/C$) to also change smoothly. Due to noise in the image measurements, the estimates of $C$ may be somewhat noisy also. In order to suppress noise we can use an approximate low-pass filter, e.g., $\overline{C}_k = \alpha C_k + (1-\alpha)C_{k-1}$ (with $0 < \alpha < 1$), to smooth the TTC output. One more register is needed to save the previous output $C_{k-1}$. The above is a simple finite impulse response (FIR) smoothing filter. To obtain better filtering we would have to remember additional old values. A more efficient approach is to use:

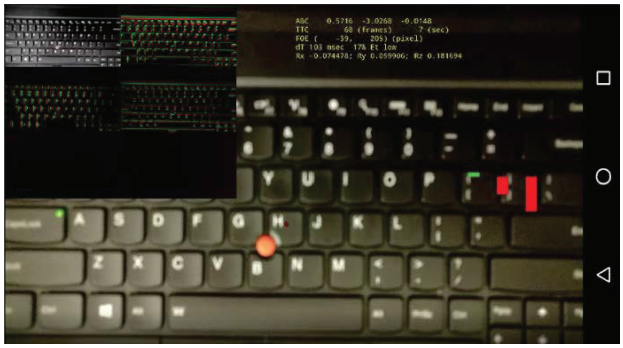
$$\overline{C}_k = \alpha C_k + (1-\alpha)\overline{C}_{k-1} \tag{37}$$

Only two registers are used. However, all previous outputs are used recursively. That is, equation (37) can be rewritten as:

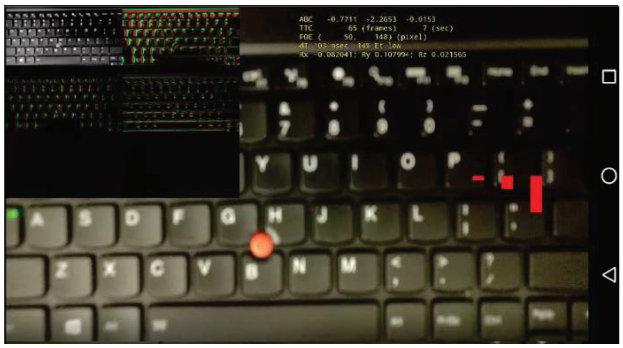
$$\overline{C}_k = (1-\alpha)^k C_0 + \sum_{l=0}^{k-1} \alpha(1-\alpha)^l C_{k-l} \tag{38}$$

This implements an infinite impulse response (IIR) smoothing filter. Here, the weights decay exponentially with time. This approach emphasises new points over old ones, and does not require keeping a complete history of old values.

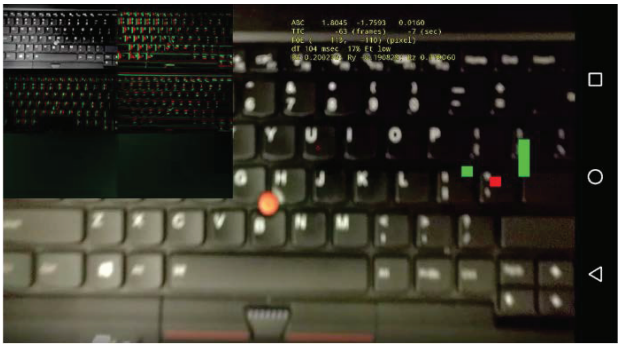
**Figure 6** Some results on Nexus 6P, (a) one frame of the results on the Android phone (b) the third red 'bar' indicates that the camera is approaching (c) the third green 'bar' indicates that the camera is leaving (see online version for colours)



(a)



(b)



(c)

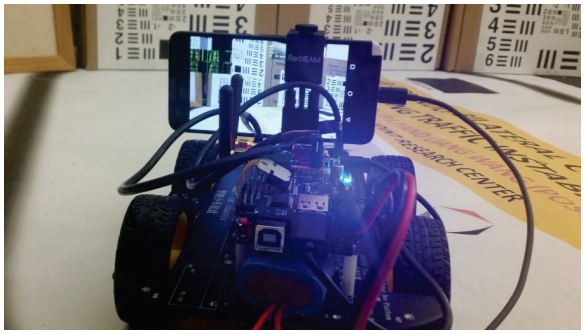
Notes: The top-left corner shows intermediate computational results. The three 'bars' indicates the motions.

## 6 Experiments

We use a controllable robot car to test the TTC algorithm. Figure 7 shows the experimental environment.

The robot is control by Arduino micro-controller. One smartphone is mounted on the robot to test TTC algorithm. Another smartphone is used to control the motion of the robot, e.g., moving forward and backward, via Wifi. The robot car is controlled to move at about 40 cm/sec. The screen of the smart phone is recorded by Android Studio IDE. The image frames are recorded by the smartphone's camera.

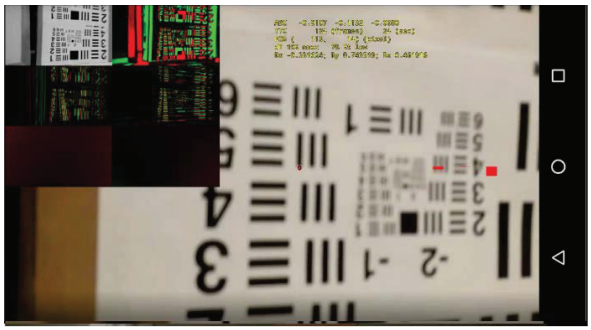
**Figure 7** The experimental environment (see online version for colours)



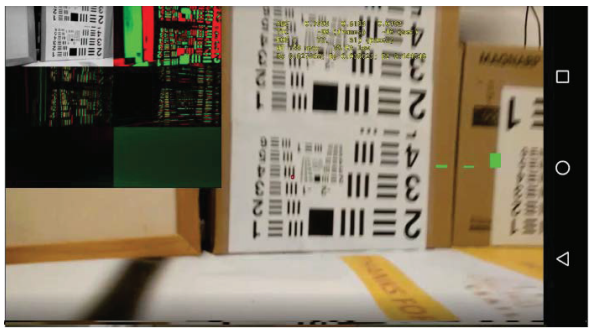
Notes: A smart phone is mounted on a controllable robot. The result on the screen is recorded on a laptop computer.

**Figure 8** The result on Nexus 6P, (a) one frame in the recorded result (b) the car is approaching the boxes (c) the car is leaving the boxes (see online version for colours)

(a)



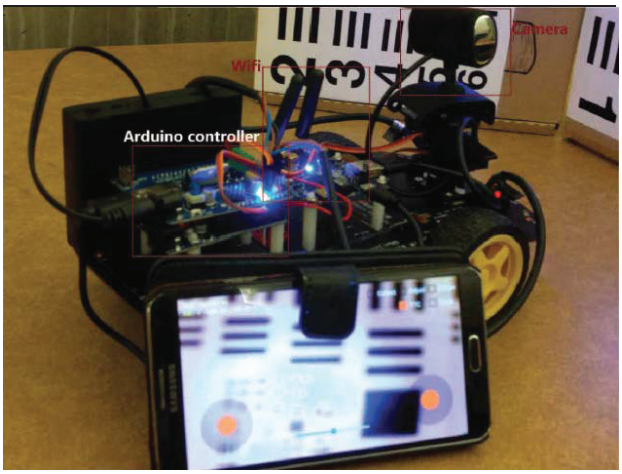
(b)



(c)

Figure 8 shows some experimental results. Figure 8(a) is one frame from the video of recorded results. Figure 8(b) [2 sec. after Figure 8(a)] is a frame when the robot was approaching the boxes. The upright bar on the right is colored red to indicate a dangerous condition (small positive TTC). Figure 8(c) [4 sec. after Figure 8(b)] is a frame when the robot was moving away from the boxes. The downward bar on the right is colored green to indicate a safe situation (negative TTC).

**Figure 9** The image frames can be recorded by an independent camera, and then be sent to smartphones for further process (see online version for colours)




The image frames used to calculated TTC can be recorded by the camera on smartphone (as shown in Figure 7), or from an independent camera. Figure 9 shows such robot. Basically, the robot contains three modules:

1 an independent camera is used to recorded the image frames

2 a Wifi chip is used to send the recorded data to smartphone

3 an micro-controller, e.g., Arduino, is used to control the robot.

Figure 10 shows the system structure. The recorded image frames (as binary data) are sent to the smartphone using Socket communication (via Wifi). The image recorded by the independent camera is down-sampled before sent to the smartphone in order to not cause noticeable delay. The smartphone can also send simple command to the Arduino micro-controller via Wifi.

The computational cost of TTC algorithm is almost negligible comparing the time used for Socket communication. In this experiment, we sent the sampling rate is set at 5 frames per second (fps), because more time is caused to send image data via Wifi than read image frames from the smartphone itself.

## 7 Conclusions

The 'stop-and-go' traffic jams on the highway are actually caused by the 'tailgating behaviour' of human drivers. We can imagine such problem can be worse when autonomous vehicles (which are designed to implement car-following control) are widely used. For safety and reliability of autonomous vehicles, the ACC system should be designed such that:

1 using larger reaction time $T$ (much larger than the sensors' response time)

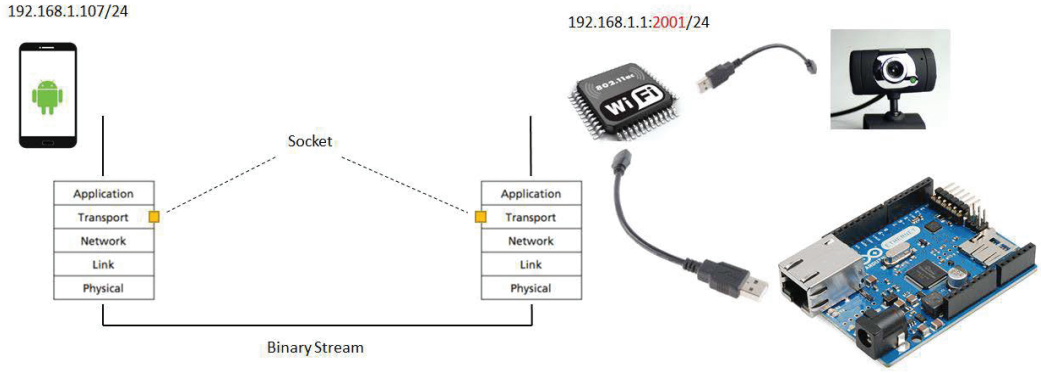
2 adding collision preventing system to avoid the inevitable potential collisions due to the traffic system's instability.

In this paper, we provide time-to-contact control to prevent the potential collisions. Simulation shows the effectiveness of TTC feedback to improve the *safety* of autonomous vehicles (see Figure 3).

For the *reliability* of ACC system of autonomous vehicles, it is *important* to estimate the relative velocity between successive cars. We could estimate the relative velocity by differentiating the distance. However, any noise in the distance measurement will be amplified by taking time derivative. Thus, relative velocity estimates by other approaches are need to generate more reliable and accurate measurements. Note that the relative speed can also be calculated by distance times 1/TTC. Thus, TTC can provide effective information to fuse with the taking-derivative-based estimate of the relative velocity, if TTC is used by another method, e.g., passive-sensor approach. Machine vision techniques provide us an effective approach to estimate the time to contact. The estimated 1/TTC can be used both as TTC feedback to fuse with the relative velocity calculated by time derivative of space measurement. Thus, the TTC control can be implemented effectively. We provide both mathematical analysis of machine vision-based TTC estimation and the detailed algorithmic implementation. The estimated TTC is given by analytical solution rather than iterative process. Thus, the algorithm is pretty fast and can even be implemented to run on Android smart phones in real time.

The 1/TTC estimated from a time-varying image is not very accurate unless considerable filtering is used (see Horn et al., 2007). However, it provides valuable information to fuse with other relative velocity measurements. Moreover, TTC provides an important criterion for the safety of cars (and robots). If the 1/TTC is large, then a potentially dangerous situation may be developing and the system may need to enter an alarm state. Vibration of the camera mounted in the car may cause the images to be smeared, which will adversely affect the quality of the estimated TTC. The gyroscope sensor in the smartphone provides a measure of the camera's rotational speed, which can be used to compensate for the image motion in the TTC calculation.

**Figure 10** The system implementation in which an independent camera is used to recorded the image frames (see online version for colours)




Note: The recorded image frames (as binary data) are sent to the smartphone using Socket communication (via Wifi).

Note that autonomous driving systems can take into account more information about the environment of the car than a human driver can. Thus, new control strategy can also be designed for autonomous vehicles by using these additional measurements 'smartly'. Horn (2013), Baran and Horn (2013), Wang et al. (2017) and Horn and Wang (2018) provide some attempts on this topic. These new designs of ACC system for self-driving cars can be optimised from two aspects:

1 yielding attractive improvements relative to today's traffic situation

2 improving the safety and performance of autonomous vehicles themselves.

TTC feedback focuses on the second objective. How to implement TTC control with the new ACC system focusing on the first objective? Will TTC feedback improve the stability of the new ACC system-based traffic, or make it worse? How to improve it? These will be our future work.

## Acknowledgments

We thank Xieyang Su, Junbo Yu, Colin Bonatti, Byung Gu Cho, Daniel Lopez Martinez, and Janille Maragh for the Android system implementation.

This work is sponsored by Toyota Research Center (under grant LP-C000765-SR).

## References

Baran, T. and Horn, B.K.P. (2013) 'A robust signal-flow architecture for cooperative vehicle density control', *International Conference on Acoustics, Speech, and Signal Processing*.

Bruss, A.R. and Horn, B.K.P. (1983) 'Passive navigation', *Computer Vision, Graphics, and Image Processing*, Vol. 21, No. 1, pp.3–20.

Chandler, R.E., Herman, R. and Montroll, E.W. (1958) 'Traffic dynamics: studies in car following', *Operations Research*, Vol. 6, No. 2, pp.165–184.

Edie, L.C. (1961) 'Car-following and steady-state theory for noncongested traffic', *Operations Research*, Vol. 9, No. 1, pp.66–76.

Herman, R., Montroll, E.W., Potts, R.B. and Rothery, R.W. (1959) 'Traffic dynamics: analysis of stability in car following', *Operations Research*, Vol. 7, No. 1, pp.86–106.

Horn, B.K.P. (1986) *Robot Vision*, MIT Press, Massachusetts.

Horn, B.K.P. (2013) 'Suppressing traffic flow instabilities', *2013 International IEEE Conference on Intelligent Transportation Systems-(ITSC)*.

Horn, B.K.P. and Schunck, B.G. (1981) 'Determining optical flow', *Artificial intelligence*, Vol. 17, Nos. 1–3, pp.185–203.

Horn, B.K.P. and Wang, L. (2018) 'Wave equation of suppressed traffic flow instabilities', *IEEE Transactions on Intelligent Transportation Systems*, September, Vol. 19, No. 9, pp.2955–2964.

Horn, B.K.P. and Weldon, E.J. (1988) 'Direct methods for recovering motion', *International Journal of Computer Vision*, Vol. 2, No. 1, pp.51–76.

Horn, B.K.P., Fang, Y. and Masaki, I. (2007) 'Time to contact relative to a planar surface', *IEEE Intelligent Vehicles Symposium*.

Lighthill, M. and Whitham, G.B. (1955) 'On kinematic waves: a theory of traffic flow on long crowded roads', *Proc. of the Royal Society – Series A*, May, Vol. 229, No. 1178, pp.317–345.

McQuirk, I.S., Horn, B.K.P., Lee, H.S. and Wyatt, J.L. (1998) 'Estimating the focus of expansion in analog VLSI', *International Journal of Computer Vision*, Vol. 28, No. 3, pp.261–277.

Newell, G.F. (1955) 'Mathematical models for freely-flowing highway traffic', *Operations Research*, May, Vol. 3, No. 2, pp.176–186.

Shahriar, N. and Horn, B.K.P. (1987) 'Direct passive navigation', *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 1, pp.168–176.

Strang, G. (2003) *Introduction to Linear Algebra*, Wellesley-Cambridge Press, Massachusetts.

Wang, L., Horn, B.K.P. and Strang, G. (2017) 'Eigenvalue and eigenvector analysis of stability for a line of traffic', *Studies in Applied Mathematics*, January, Vol. 138, No. 1, pp.103–132.