

# The Tangible Car – Rapid Intuitive Traffic Scenario Generation in a Hybrid Table-top and Virtual Environment

Marcus Tönnis\*

Technische Universität München

Fakultät für Informatik

Boltzmannstraße 3

85748 Garching b. München, Germany

Tel/Fax: +49 89 289 17083 /-17059

## ABSTRACT

Driving simulators are still a worthwhile environment in the development cycle of in-vehicle information systems. In general, traffic scenarios are somehow artificial or depend on a certain application logic controlling foreign cars relative to the own car. To generate traffic scenarios with more human behavior content, an intuitive system to enable traffic scenario development for user study subjects has been developed. The system uses a tangible miniature car in a table-top environment that is connected to a driving simulator. This system enables analysis of human behavior to facilitate the development of semi-automated and especially driver accepted in-car assistance systems. System architecture and experiences from monitoring people using the system while generating traffic scenarios are reported.

**Index Terms:** H.1.2 [Models and Principles]: User/Machine Systems—Human Information Processing; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Ergonomics;

## 1 INTRODUCTION

For the development of semi-automated cruise- and course control driver assistance systems it is not only necessary to ensure and enhance safety, but also to care for driver acceptance. The Adaptive Cruise Control (ACC, [1]), for instance, is such a system. It supervises the distance to a leading car and adjusts the own cars speed, so that a safe distance is kept. When, for instance, a leading car is slower than the own car, the automatic distance control system starts decreasing speed at a certain distance. Sportive drivers may estimate the distance too high and the relative speed difference too low, thus appraising the system's early speed decreasing activity as annoying. In contrast, cautious drivers may often get nervous, if the system reacts too late in regard to their personal habits. People then tend to turn off the assistance system, because it does not fit to their own behavior. Thus all safety aspects of the assistance system get lost.

Future assistance systems will not only monitor longitudinal behavior, but lateral and environmental settings as well. Hence the problem of acceptance gets even worse. To test such system's impact on personal habits it is, in common, required to deploy the system to a real car and to test it on traffic restricted test drive road courses. This is an expensive and time-taking process.

To enable evaluation and insight about personal behavior at an early stage of development, a system for laboratory environments is required to generate traffic scenarios. Computer generated traffic settings can not reflect all the minimal occurrences in traffic behavior, as human generated trajectories do. Driver's often react on

minor changes in another car's trajectory or speed. To give an example, one may remember a personally experienced traffic situation, where the driver of another car did nothing special except for releasing his gas pedal when approaching a branching road. Not even seeing brake-lights or the turn signal, one is sure that the other car will do a turn into the other road. A second well known scenario takes place on multi-lane roads, when one sees a car slightly approaching its own lane's border. Somehow one knows that the car is going to do a lane change. A human-controlled traffic generation system enables various opportunities for evaluation of such assistance systems. An environment to illustrate and discuss traffic scenarios for psychologists and sensory researchers enables life experience. In advance such a lab system enables test subjects to easily generate traffic scenarios on their own. Then the delta between the desired assistance system's and a human's reaction can get measured by putting other test subjects into a interconnected driving simulator to experience different setups.

Such a system has been built, incorporating a *table-top back-projection workbench* with a *tangible miniature car* (see Fig. 1) into a *driving simulator virtual environment* (VE). Here a live system enables collaborative discussion, easy intuitive and rapid traffic scenario generation and multi-view experience. To describe the system and its application, first, the problem of a comparison between assistance system behavior and human behavior is introduced to motivate the application of the hybrid system. This problem leads to the issue of how to generate human behavior, discussed subsequently. Related approaches are illustrated and, in advance, the new solution, a human controlled tangible car in a table-top environment is discussed. After showing how the system incorporates in a driving-simulator environment, certain issues and experiences for people dealing with the system are discussed. The human related issues concern object calibration of new virtual 3D car models and scenario generation on a fixed size table-top.

## 2 TRAFFIC SCENARIOS

Traffic scenarios are certain occurrences of a special situation. Such scenarios play a major role in the development of driver assistance systems as they state the baseline for concepts to react on traffic behavior. This section illustrates, how scenarios differ and why Advanced Driver Assistance Systems (ADAS) have to adapt and react on these differences. In advance, this section explain the necessity of repeatable scenarios in user studies and discusses approaches to generate traffic scenarios.

### 2.1 The Delta between Human and Assistance System Behavior

A car driver on the road has to react on many visual stimuli from the surrounding environment to maneuver safely. All visual stimuli are necessary input to control the car correctly, to keep in on the road and not to hit any obstacle. ADAS systems support driver's in

\*e-mail: toennis@in.tum.de

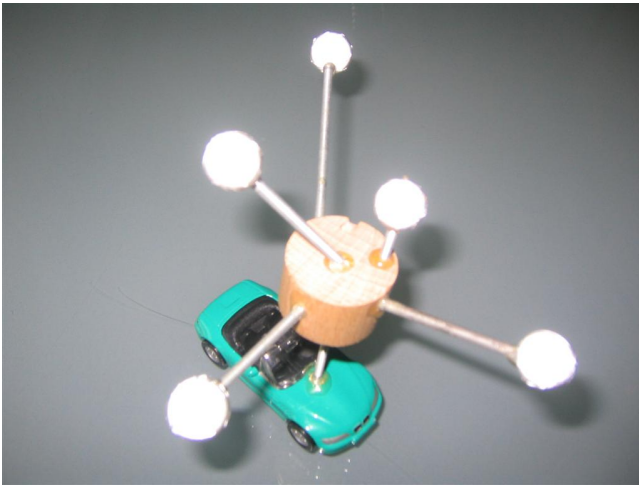


Figure 1: The tangible miniature car: A small green BMW Z3 series in a scale of 1:100 with attached 6 dof marker tree.

the monitoring task and inform the driver about insecure maneuvering activities or overtake certain responsibilities of the driving task. Thus ADAS systems are active components to increase active and passive safety. If, as explained earlier, such an ADAS system does not adopt sufficiently to a driver's personal behavior the risk of the system to get turned off increases. In-car systems that are turned off in general burden a driver's comfort because they annoy the driver, or because the driver is uncertain about the system's behavior in traffic or because the system is boring to the joy of traveling. Thus all intended aspects of enhanced safety get lost as well.

To keep ADAS systems active, sufficient and valuable rules for the system's behavior need to be defined. Such rules derive from the input, a car driver is exhibited. Usually the rules must derive from visual stimuli, a driver is exhibited to, especially for an ADAS system's output. Rules not matching to driver intrinsic rules, generate the so called *Delta* in behavior.

Surely every human has another personal habit in behavior. Some prefer to drive more active than others or want to have larger distances between cars than required for safety. ADAS systems must adapt on those personal differences specifically in future. A certain baseline enforced by traffic law and a general human feeling for safety has to get reached. To differ between personal and general human behavior, factors contributing to behavior need to be analyzed. Other car's trajectories play a major role here, but there are various additional factors, such as the type of the other cars, their relative speed, usage of turn signals, etc. To analyze which factors are relevant input for a driver's behavior and which are not, user studies are required where single factors can be altered. Only factors that generate a significant difference in behavior are useful for rules in ADAS systems.

To determine sufficient and valuable rule sets for driver-accepted ADAS systems, test subjects need to be exhibited to traffic scenarios, where single factors are modifiable. Repeated test drives where, the driver controlled trajectory and the ADAS system's (intended) actions allow determination of differences and thus the delta. To develop behavior schemes for future assistance systems it is necessary to compare these differences in a driver's personal behavior in traffic to adjustments of the automated driving assistance system.

## 2.2 Repeatability of Traffic Scenarios

Repeatability of traffic scenarios is an important issue for user studies. Only when a certain situation is fully and clearly repeatable it

is possible to use it in user studies. Certain factors can then be separately be altered. A user study to distinguish significant differences can be performed. If more than one factor at a time is changed, results can not clearly be attributed to the originating cause. Thus traffic scenarios require explicit repeatability to compare one to each other.

It is possible to gain insights about human behavior for comparison in two ways: First, test persons can get exhibited to real traffic. Second, simulated traffic in a driving simulator can get used to gather data from the scenarios. Both approaches enable to compare reactions of the assistance system and the driver. By recording the driver's actions as well as the actions, a ADAS system would perform, based on its rules, both behaviors can get analyzed and compared.

Both approaches require previous definition of certain scenarios. Such certain scenarios can, for instance, be given as follows: A three-laned highway, the own car on the middle lane, another car on the right lane, going to do a lane change to the left. A certain reaction is necessary to avoid a crash. The expected difference in behavior between a real driver and the system could be a lane change to the left lane instead of braking.

The first approach to determine human differences uses traffic scenarios from real traffic. Recording the behavior of existing or testable assistance systems and humans in general uses in-car bus systems as the CAN-bus. An additional camera is then used to record the driver's field of view. Road courses are driven until a certain predefined scenario occurs. This point in time is marked in the records and the movie. Later, recorded CAN-bus data from the assistance system is compared to recorded values from steering wheel and pedals. Analysis can, for instance, show that the ADAS system reacted much earlier and decreased speed when approaching a slower car from behind. The early reaction can just be annoying. Braking events, where the ADAS system generates false positives are also possible. This can occur in curves, where another car is on another lane, but, depending on the curve get directly in front of the own car's sensor system. Here depending on distance and differential speed, the system could start braking or turn itself off, requesting the driver to take over control again. Both examples were taken out of analysis, where an existing ACC system has been compared to natural human behavior.

The approach to monitor real traffic allows for analysis of general reactions, but does not allow to compare groups of scenarios where general reaction schemes are to be analyzed. Even if the same scenario occurs again, there is always more than one small difference, at least in the trajectories of the cars, prohibiting exact comparison to other scenarios.

The non-comparability of the real-street setup enforces to move into simulated environments, where traffic scenarios are repeatable with adjustable single factors. Trajectories of cars can be generated as absolute movements or as movements relative to the own car. In advance, passing certain points on the road can trigger certain activities and behavior of other cars, e.g., to brake abruptly. Further extrinsic factors as different car types and colors are adjustable as well. Scenarios are repeatable as often as required. Various factors in the setup can be modified and general rules for ADAS system behavior can be deduced.

Here the benefit of computer systems enables comparison of scenarios, not possible in real traffic. In contrast, computer generated trajectories are exact. A simulated car always follows a perfect course, in general interpolated between predefined points on the road. Driver's often react on even minor changes in another car's trajectory or speed. Such minimal changes in behavior are *somehow* noticeable for humans. These are difficult to generate with simulation systems and never reflect natural behavior. The minimal differences between simulated traffic and real traffic are necessary elements of a driver's visual input.

Thus it is not only necessary to test subjects through behavior in traffic scenarios, but to include people into the design of traffic scenarios. Only through this early integration, traffic scenarios reflect not only pure computer generated road-courses, but also required human inexactness.

### 2.3 Test Subject related Scenario Development

In general, people are tested in driving simulators to collect data about interaction or reaction to new in-car systems. To include people into the development of a traffic scenario is a new way of participation in test trials.

Again, there are two opportunities to generate traffic scenarios. The first opportunity just requests people to drive a real car, record car and sensor data and to reuse it in in virtual driving-simulator environments. The second possible way to result in traffic scenarios is to instruct participants to a certain general setup and to let them develop that scenario.

The first approach to reuse sensor data eases test participants life as it only requires them to drive a car as under normal conditions. The car's position, speed and steering activity is recorded. Sensor data, e.g., from an ACC sensor is recorded as well and used to overlay tracked positions by simulated virtual cars. This procedure has been executed through use of sensor data from an second generation ACC sensor. This sensor tracks up to 32 objects and sorts them by relevance. The most important object is labeled as ID 1 and subsequent objects follow. If another object in the field of sight of the sensor becomes more relevant, IDs are changed. This approach is useful for the application in an ACC system, but complicates reuse in driving simulators as associated virtual models of cars would spontaneously jump to another position. Also environmentally fixed objects can get tracked by the ACC sensor and thus become cars. Fig. 2 shows two snapshots to real traffic that has been put into a virtual world. Fig. 2(a) shows some cars. At first sight this scenery looks normal, but on a second glance, one may realize some overlapping cars. This occurs, if an object produces more then one spot tracked by the distance sensor. In Fig. 2(b) other objects than cars have been tracked. A look on the corresponding camera image revealed a guardrail to the left side of the own car.



(a) Some cars generate more then one reflection, thus leading to two overlapping cars



(b) Many cars on the left side. Here reflecting a guardrail

Figure 2: ACC sensor data used to generate traffic scenarios without any filtering. Obstacles in the environment are treated as cars

It probably would be possible to analyze the sensor data, identify environmentally fixed objects and to attribute unique IDs to all remaining tracked data. If an object gets lost from tracking, a simulated car would stop or have to follow a computer generated trajectory again. If this car then is tracked again, the risk of an overlapping computer generated and new tracked trajectory comes up.

The second approach puts more load on the test participants. Here certain predefined scenario boundary conditions are given to the test subjects and they are requested to build their own scenario

in a virtual driving simulator environment. Collecting several solutions for the same given boundary conditions allows for analysis of similarities and differences in behavior. As test subjects are normally not requested to build traffic scenarios on their own, tools to support this activity vary in practicability. Non computer-skilled people have to use traffic development tool, which often requires them to handle with mathematical equations and sort of programming.

The easiest opportunity for development of traffic scenarios is general programming. Here test participants would need to learn the corresponding programming language, which apparently is not possible for every test person. Applied tools as OpenFlight [9] and Dynaware [11] include sophisticated driving dynamics, traffic models and even sensor simulation environments. They apply for testing of ADAS systems and provide predefined procedures for certain maneuvers. These procedures base on mathematical equations and always reflect an exact route. Every maneuver can be integrated into a whole drive and can be overlaid with any type of car, even new 3D models. A similar approach is used in the Iowa driving simulator [2]. Here dynamic and behavior models are separated and use a state machine to control dependencies between different vehicles. The Iowa system allows to focus on behavior modeling without taking care of a realistic driving behavior. Still the development of scenarios is a time-consuming process and never will reflect a test person's manner. Finally, those systems are complicated to distribute onto complex architectures using more than one computer. Cars on one computer would not know about cars controlled by a second computer.

### 3 RELATED WORK

To generate trajectories for diving simulator experiments, research concentrates on analysis of sensor data.

Darms [3], for instance, developed a system to deduce data for a Lidar Laserscanner. Resulting data sets contains points including speed.

Similar approaches were under examination by Lindl et al [12]. They combined several sensors in a car to get more data from out of the environment. Their work focuses on sensor-fusion and detection of different types of road users.

Kirchner et al [6] and also Polychronopoulos et al [8] use models to detect obstacles on road. In advance, they also detect the road-course.

Data from all these investigations can be used in a driving simulator to animate cars. Quality of these systems, based on alternative sensors and sensor fusion outperforms sensors, available at the moment and thus enables better and more realistic reuse of real traffic occurrences.

Other approaches try to generate car movements in laboratory setups. The Tangible Pathfinder [10] is a TUI based orientation and mobility trainer for visual impair. The Tangible Pathfinder is intended to allow detailed, autonomous learning of a new physical setting and self-assessment of the resulting cognitive map. It allows for gaining information about objects and route layouts by touch but is not intended to generate traffic models.

Kanec et al [5] use a physical car model, moved by hand to maintain a virtual reality representation of the environment and to use it for producing additional views and to provide guidance to users in a parking situation.

Novak et al [7] use a miniature car in a table-top environment to generate a drivers personal view. The table-top environment is a bird's eye map, enabling test coordinators to move the miniature car through a city environments. Their system allows for evaluation of attentive in-car user interfaces.

## 4 CONCEPT - A TABLE-TOP TANGIBLE CAR

The concept developed in my work uses a tangible miniature car to generate car trajectories for driving simulator virtual environments. To give a view on the environment, the car is in, not only the driving simulator's driver field of view is presented, but also a table-top bird's eye view of the near environment of the car.

This approach keeps some metaphors of human behavior intact and allows for fast and intuitive traffic scenario development. To illustrate, how this concept incorporates in tangible interaction concepts, first reflection of application metaphors are discussed in general, followed by illustration of transported metaphors, still alive in the table-top environment.

### 4.1 Metaphors of Tangible User Interfaces

Tangible user interfaces [4] reuse real world metaphors in augmented or virtual environments. In general, one metaphor is selected and extended into the new application domain. In our concept the main metaphor of the car is to move through the environment. Moving the car directly affects the table-top presentation, where a virtual representation, a virtual car model moves around. The driver's view also is affected directly. People sitting in the driving simulator immediately get the visual feedback as if they were driving. Working at the table-top environment allows to perceive both views simultaneously, thus facilitating front view as well as environmental experience.

Often, not only one metaphor is content of any TUI - other metaphors are transported as well. For instance, a pen's use is to write, but often it is used to point. When reusing a user interface in extended environments, one has to decide, if other kind of interaction is intended or not. The extra metaphors can be explicit or implicit. Implicit meaning is transport whether the developer intends it or not. All non-intended metaphors contribute to a certain delta in usage and can not get turned off.

Thus when designing and evaluating such tangible devices, drawbacks in usage can be achieved, but it is difficult to monitor the source of the drawback, because it is unclear, which factor, which metaphor is the source of the drawback. One only can try to collect all transported metaphors and to distinguish, which are useful and which are not.

### 4.2 Corresponding Metaphors

The tangible miniature car's main intention is to reflect a real car. Already children playing with cars move them almost similar to real cars. They seldom move them sideways, mostly to naively simulate future parking systems. The main use is to drive them through the environment, and in children's case to make races and accidents. This takes place in a straight forward or curved manner, similar to real cars. Except for producing accidents miniature cars are promising for use in VE.

Interaction affected by such a miniature car is enhanced by a concurrent generation of a personal view, as if the driver would be sitting in a driving simulator. Driving there and seeing another car spontaneously appearing and quickly approaching can have an effect of surprise. This leads to a reflex, in general, drawing aside. This reaction is similar to a driver's action to avoid a crash in a dangerous situation. One can imagine the situation, when one forgot to eye on the back-mirror or missed to check the dead-angle, starts to do a lane change and then realizes another car on that lane. The driver is alarmed and immediately pulls back to his own lane. The pull-back action is given by a immediate shaking with both hands on the steering wheel. The steering reaction is transformed to a shaking action when using the miniature car.

Collaborative experience and discussion also is enabled by this concept. As every virtual car has a real representation, it will not possible to move a second, later integrated car at exactly the same

position. Collisions are producible, but complete overlappings will be prohibited.

Not only a single car's behavior, but also complex traffic behavior can be generated. After a first car has been recorded using the tangible miniature car, a second trajectory can be recorded by associating the real to a second virtual model. Here in fact, overlappings are possible, but in general will be avoided, as adult people participate in experiments and are not requested to play.

## 5 SYSTEM SETUP AND ARCHITECTURE

The system incorporates in a driving-simulator environment and a table-top environment into a hybrid multi-view setup. Fig. 3 shows the setup in the laboratory. One part of the laboratory is equipped with a driving simulator, consisting of a projection wall with a 40 degree field of view and a driver's cockpit on an aluminum frame (see Fig. 4). This setup enables life experience from out of the drivers perspective. The driver's perspective is visible from the second part of the laboratory, containing the development and discussion environment. The second part consists of a table-top environment, placed in an infrared tracking volume (see Fig. 5). The tangible miniature car (see Fig. 1) can be placed on that table, while the road scenery is projected onto the workbench.

## 6 OBJECT CALIBRATION

The system allows test participants to bring in their own, preferred virtual car models. Every major format is accepted by the system. To effectively use these models, a user has to calibrate the virtual model to the 3D car. This procedure should be performed by the user of the system and not by the test coordinator. If the test coordinator would perform this activity, the test participant would be moved out of the system for the meantime and could feel superfluous. It is necessary to keep the tested person in the system to communicate a feeling for use of the system and to later enable him to generate scenarios. In advance, repeated disruption through the test coordinator would make a test subject feel uncomfortable, which can decrease the quality of the resulting scenario.

Thus a easy to use calibration method was required in the system.

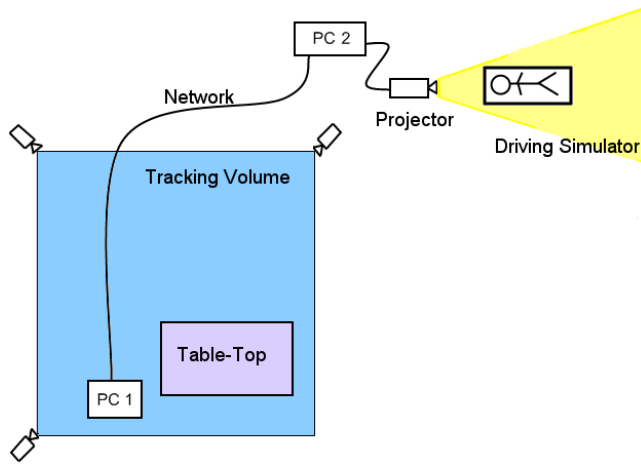
### 6.1 Object Scaling

The first decision to make was about scaling. A newly integrated 3D virtual car model had to be scaled to fit into the environment. Its size had to fit to the road model and to the size of the tangible miniature car. Here a simple slider has been integrated in the central controlling GUI.

Test participants had no problem to deal with that slider and easily could adjust the size of the 3D model. An interesting observation about the relationship between lane size, 3D model size and the size of the miniature car could be monitored. During the first system usability tests, not even the road size (lane size) was predefined. Test subjects could adjust all sizes of virtual models in dependence to the miniature car. In the first trials, the size of the virtual 3D model was accurately adjusted to the size of the miniature car (see Fig. 6(a)), generating the effect, that the virtual model was nearly occluded completely. Thus the virtual 3D model appeared unnecessary for scenario development. After adjusting the size of the virtual car, the road section had been scaled to fit to the size relationship of the real and the virtual car, making the road relative small. The visible road section on the table-top environment got a length of about 70 m (in real life).

In this setup, test participants could use the system, but generated higher lateral errors in the trajectory through shaking the car when they moved it around and could not interpret the virtual cars movement, even if the real car directly gave position and orientation.

In further tests the road section had been made wider, reducing the length to 40 m. Now, the miniature car was about 60 % the size



(a) Sketch of the setup



(b) Photography of the laboratory. The table-top environment normally would be more to the left, marginally behind the driver's cockpit

Figure 3: The system setup: The table-top environment in a tracking volume, where the tangible car can be moved around. The system is connected to a driving simulator

as it should be to fit in the lane. A virtual car model integrated into the scene now was about 1.6 times larger than the real car (see Fig. 6(b)). Now test participants could directly see the virtual representant of the miniature car and perceive the transformation they made in the virtual world.

This miscalibration of both relationships eased use of the system. Moving errors of the real car had less impact on jitter in the virtual world. Perception of the virtual representant gave better understanding of the behavior of the virtual world and test participants felt more comfortable when using the system.

## 6.2 Object Alignment

The second and more complex activity is to correctly align the virtual model to the real 3D model. Approaches to use GUI-based manual tracking facilities in general fail, as scene-graph-based 3D transformations are not easy to understand and to deal with by use of 2D input devices like mice and keyboards. An easy to use calibration algorithm has been implemented to enable every kind of user to align their 3D models correctly.

The final calibration method just requested a user to place the



Figure 4: The driver's cockpit is build on an aluminum frame and has the same relationships as a common driver's cockpit (distances and sitting height)

miniature car at a certain position. The user can decide, where to place the car. It only is useful to align it the some other object, so that this position and orientation can not be forgotten (see. Fig. 7(a)). After a click to store the current position, the virtual model has to get aligned to the previous position of the real model (see. Fig. 7(b)). A second mouse-click to store the virtual models transformation activates the calibration routine. Here both stored transformations are composed (Composite Model Transformation) to result in the final calibration transformation. The result is a good aligned representation of the 3D virtual object to the real miniature car (see. Fig. 7(c)).

Working on a table-top environment in a bird's eye view can lead to misalignments in vertical alignment. Here a gravity algorithm enforces the virtual object to „fall“ to the horizontal plane and to tilt until three points of the lower end touch the horizontal plane. The resulting offset is stored, so that no second calibration is needed, if the 3D model is used again to simulate a second car.

The calibration algorithm has been tested with some people to guarantee applicability of the algorithm. The principle of the algorithm is shown in the controlling GUI and people could easily deal with that subsystem. Only associating the tracked car with the virtual model sometimes confused test participants, as they wondered, why the virtual model immediately jumps to the position of the (now) tracked real car.

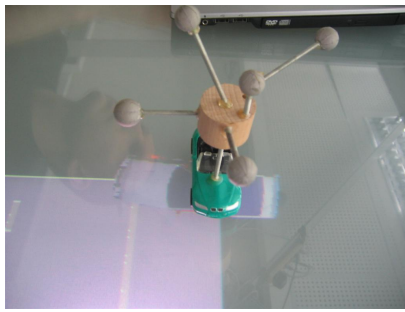
## 7 SCENARIO GENERATION

Generating a scenario is an iterative process, where a car's trajectory is recorded one after another. After a first trajectory has been recorded, a second car is incorporated and its trajectory can be recorded in dependence to the course of the first car. All other cars movements are recorded subsequently.

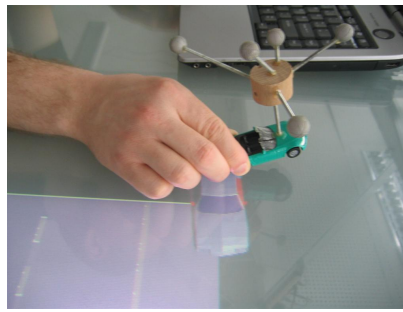
To effectively deal with the system and the available space, certain further issues arose in the development. Issues as the limited range of the table-top environment are discussed subsequently. The following sections discuss these issues and illustrate experiences, people made, when dealing with the system.

### 7.1 Controlling the Tangible Car

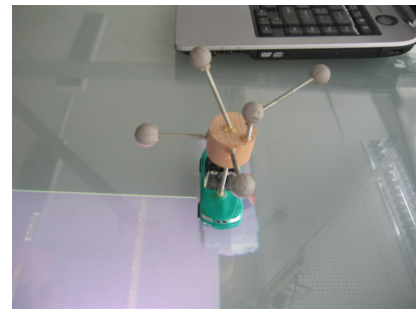
The whole setup in the laboratory enabled users of the table-top environment to additionally monitor the driving simulator view. Thus both perspectives helped people to perceive changes made to the



(a) Placement of the real miniature car



(b) Manual alignment of the virtual model



(c) Resulting calibration

Figure 7: Calibration of a new 3D model to the tangible car

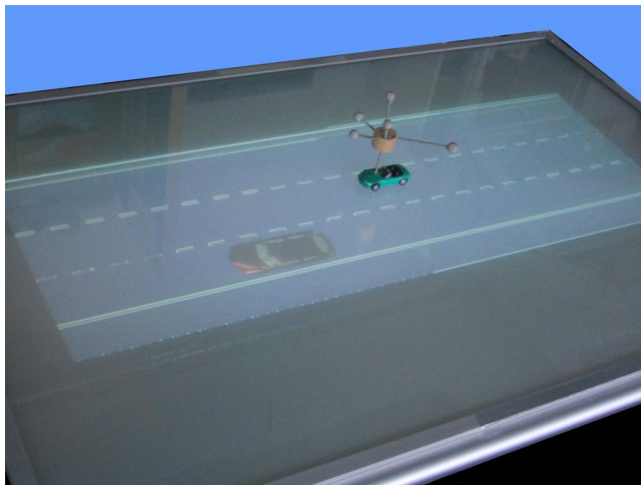


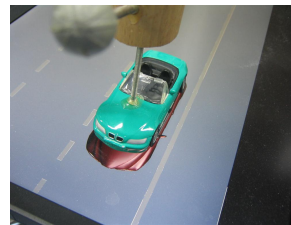
Figure 5: The table-top environment is built with a back-projection workbench and is placed in an infrared tracking volume

controlled car's trajectory. Orientation of both views followed the same orientation: pushing the tangible car in a forward direction changed the driving simulator's view in the equal manner, making the driver's view go into the same direction in the laboratory.

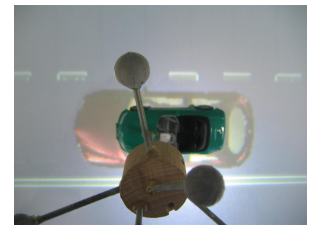
People can control the car from every side of the table-top environment. In general, right-handed people preferred to use their right hand and left-handed people used their left hand to control the car (see Fig. 8). Few people tried to stand on the small side of the table: Controlling the car from the side, the cars drive to, immediately was neglected, because the driving simulator's view then was in the user's back. Controlling the car from the other small side, where the car's came from, allowing people to get a view similar to a third person view as in 3d-action games quickly was given up, because it became difficult to reach over the whole table (length over all: 1.1 m). Thus the tangible car was controlled from „besides“ the scenery, allowing people to stand facing in the same direction as if they would sit in the driving simulator.

## 7.2 Iterative Recording

To then generate traffic scenarios, a user can use predefined road setups. A road model is placed in the table-top and in the driving simulator. The generation starts with loading and calibrating a 3D virtual model. After that, the real car movements control the virtual representation. Simply pushing a record button stores every transformation of the real car in a *PoseStream*. After the recording has



(a) Photography of a 1:1 scaling



(b) Photography of a scaling with not equal size. The virtual car model is about 1.6 times the size of the miniature car

Figure 6: Different scalings

been stopped, the *PoseStream* is stored and associated with the 3D model of the car as its data-source. Now another 3D model can get associated with the car. Replaying the scenario and concurrently recording a second car's trajectory with the newly assigned 3D model enables generation of complex traffic scenarios. A screenshot taken from a scenario, where a double passing situation using all three highway lanes occurs is given in Fig. 9

The iterative approach to develop traffic scenarios was easily to understand and apply for the test participants. After calibrating their first car, they immediately could start and record the trajectory. People then had to reassign the tracking system to now track a second car. As the system at the moment relies on a GUI to control the system itself, people then had to activate scenario playback and recording concurrently. Here a security mechanism had to be integrated after first tests to prohibit that a virtual model that has a *PoseStream* as data-source can also have the tangible car as input. Not having that mechanism, often resulted in people wondering why the car continuously jumps between two positions.

Test participants in general used the table-top environment to generate a new *PoseStream* and then used the driving simulator view to check the recorded path. In the beginning, people often were not satisfied with the route, the recorded car took and re-recorded to path. After a while, when people got familiar with behavior of the miniature car on the virtual road, the first trajectory was good enough to face test participants wishes.

Only the step from the system control GUI, where the test participants had to activate and stop the recording, disturbed them, because they had to turn from the computer to the table. Future extensions will have to add a record-stop-facility on the table.

## 7.3 Limited Interaction Space

The used table-top environment had a width of about 1 m to 0.7 m. The road course, a three laned highway, used about half the width

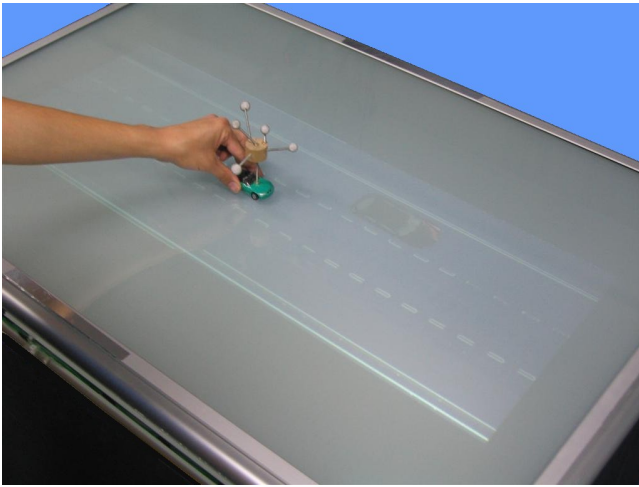


Figure 8: The tangible miniature car can be controlled from every size of the table allowing test participants to get into their preferred maneuvering position

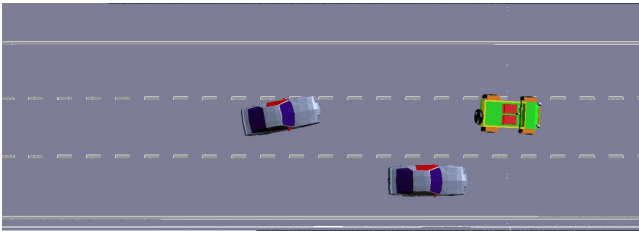


Figure 9: A screen-shot of a scenario: a car (left) is going to pass another car (right) that just passed a third car (middle)

of the projection area, thus having a real size length of about 40 m. This range only would allow for development of parking scenarios.

To meet this issue, the bird's eye camera also could be attached to a PoseStream. One PoseStream was originally provided with the system. At a speed of 80 km/h, it leads straight along the highway. If this PoseStream (or any other later generated PoseStream) is attached as a source to the camera, the whole scenery becomes animated on replay of a scenario. Within the system, the PoseStream recording facility composes the position of the camera in the scene and the position of the tracked car on the table-top. Thus every movement of the tangible car reflects the correct position on the moving ground.

Fig. 10 shows the part of the whole dataflow in the system, responsible for this composition: The Camera PoseStream (top left) controls the Camera (right). For the camera, a static Offset (middle left) is fused (two Fusion Handlers) into the resulting transformation. Tracking data (DTrackPoseSource, lower left) is transformed to the system's scale (DTrackToParentTrafo) and fused (two Fusion Handlers) with the Kamera PoseStream. The fusion is performed by the CameraZTrafo (deprecated name) and provides the Pose of the Controlled Virtual Car.

Given road courses with curves, the remaining free space on the table-top can get used to allow for curved scenarios by use of the one given straight PoseStream. To use not only a straight line through the environment, further PoseStream (at best associated to the road course) should be predefined. Alternatively, already now every road course can get used in the system. One only has to record

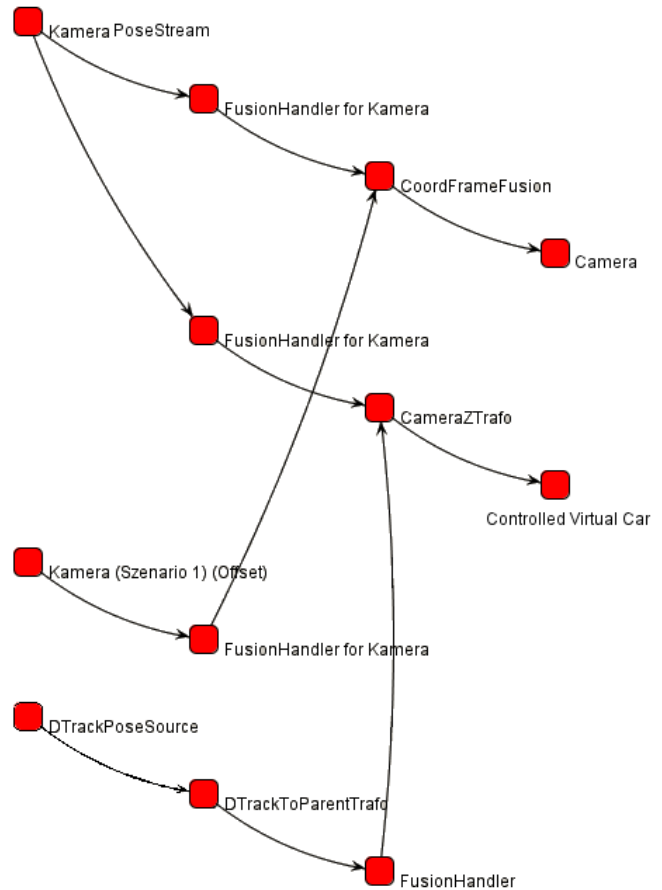


Figure 10: The part of the dataflow network responsible for the base speed of the tangible car

a curved PoseStream and then attach in to the bird's eye camera. From now on the camera will follow this trajectory. Repeated new recording and attaching to the camera can reach every point on the simulated surface.

Test participants again understood the concept of the flying camera and intuitively knew, that their car now has a base speed, the speed of the camera above the scenery. Moving the car forward did increase the car's relative speed and moving it backwards decreased the relative speed. These facts were understandable to all test participants and, especially in highway scenarios, they could efficiently develop any kind of scenario. Further scenarios to deal with crossings will definitely require suitable camera-PoseStreams to allow for useful anticipation of the crossing interval of the road. These PoseStream will have to follow a general crossing-speed-behavior, decreasing speed when approaching the crossing and a later increase of speed after the crossing.

Here by use of an animated camera, the turn-over from the computer to start and stop the recording mode generated delayed reactions of test participants, because they immediately had to take over control of the miniature car. Right after clicking the record button on the GUI, the car started driving at the camera's base speed. Introduction of a delayed activation or a table-top controlled start-stop-mechanism will solve this issue.

Although the system only does incorporate one tangible car at the moment, collaborative work is enabled. Various PoseStreams can be recorded per car. Traffic scenarios then can be altered by dynamically associating different trajectories to the same or differ-

ent cars. Teams of psychologists can discuss about human behavior and illustrate behavior to developers of sensors. They can explain, where and to what extend a sensor has to track certain areas that are more important to supervise than others.

Further integration of more than one tangible car can have a wider impact on scenario development and team discussion. Each participant can control his own car and can contribute to a depending traffic behavior.

## 8 CONCLUSION AND FUTURE WORK

A system to intuitively and easily develop human behavior based traffic scenarios has been illustrated. The system allows to collaboratively work in a hybrid table-top VE. It can deal with various high quality graphic models and textures



Figure 11: A screen-shot taken from the driver's perspective. The system allows for high quality virtual models and textures

Test participants easily understood, how to generate scenarios, but sometimes had problems with using the GUI. Recording a trajectory was easy for any tested subject. It took some recording, to get a feeling for realistic rotation angles when changing lanes. Trained people can generate a scenario including five cars on a one minute highway course in about 8 minutes, excluding object calibration, because all models already had been used and were calibrated. Every recording of a trajectory takes as long as it takes to drive that trajectory. Only some seconds on the controlling GUI extend the time to generate a scenario.

The system allows for intuitive and rapid development of traffic scenarios. It further enables a new field of research, where test subjects can get integrated into a human behavior analysis. Collaborative aspects are met, e.g., in discussion, about „when should a system react how“ or discussing by multiple tangible miniature cars

Future Work focuses on integration of a driving dynamics model to correct pose streams after recording. Correction is necessary because humans tend to turn the miniature car too much, thus generating too extreme view turns in the driving simulators field of view. At the moment the system only supports car (and camera) trajectories. To have a system that can provide a more realistic world, also additional visual stimuli will get integrated, such as turn-signal lights. Finally, the driving simulator's pedals and steering wheel will be connected to system to enable round-trip scenario development. Then we will bring the system in a user study to completely evaluate its facilitation for rapid traffic scenario development and transmission effects of human behavior factors.

## ACKNOWLEDGEMENTS

I like to thank BMW Forschung & Technik GmbH for funding of research and for valuable and enlightening discussion. I furthermore

want to thank Marek Orschewski for his efforts in implementing parts of the system, testing it with non-computer related persons and for making it usable for non-computer-scientists.

## REFERENCES

- [1] BMW. ACC. <http://www.bmw.co.za/products/acc/default.asp>.
- [2] J. Cremer, J. Kearney, Y. Papelis, and R. Romano. The software architecture for scenario control in the iowa driving simulator. In *Proceedings of the Coliference oil Computer Gellerated Forces and Behavioral Representation*, Orlando, Florida, 1994.
- [3] M. Darms. Eine Systemarchitektur zur Fusion von Umfelddaten. Mar. 2004. Aktive Sicherheit durch Fahrerassistenzsysteme, VDI, Germany.
- [4] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA, 1997. ACM.
- [5] K. Kanev, N. Mirenkov, and A. Urata. Parking Simulation and Guidance in a Model Environment.
- [6] A. Kirchner and T. Heinrich. Model based detection of road boundaries with a laser scanner. *Proc. Int. Conf. Intelligent Vehicles*, pages 93–98, 1998.
- [7] V. Novak, C. Sandor, and G. Klinker. An ar workbench for experimenting with attentive user interfaces. In *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*, Arlington, VA, USA, Nov. 2004.
- [8] A. Polychronopoulos, U. Scheunert, and F. Tango. Centralized data fusion for obstacle and road borders tracking in a collision warning system. *Proc. 7th International Conference on Information Fusion, Stockholm, Sweden, 2004*.
- [9] Presagis. OpenFlight. <http://www.multigen.com/products/standards/openflight/index.shtml>, accessed 20.09.2007.
- [10] E. Sharlin, B. Watson, Y. Kitamura, D. Rorabeck, R. Lederer, S. Sutphen, M. Takimoto, and F. Kishino. The Tangible Pathfinder Design of a Wayfinding Trainer for the Visually Impaired. *Proc. Graphics Interface, 2004, 2004*.
- [11] TESIS. DYNAware. <http://www.thesis.de/de/index.php?page=583>, accessed 20.09.2007.
- [12] L. Walchshausl, R. Lindl, K. Vogel, and T. Tatschke. Detection of road users in fused sensor data treams for collision mitigation. In *Proceedings of the 10th International Forum on Advanced Microsystems for Automotive Applications (AMAA'06)*, Berlin, April 2006. VDI/VDE/IT.

**Marcus Tönnis** Dipl.-Inf. Marcus Tönnis has studied physics and computer science at the Technical University of Munich, focusing on software engineering and data management. The topic of his diploma thesis covered the development of a data management architecture for ubiquitous ad-hoc networks and Augmented Reality. He spent a year at the software company Beck et al. before joining Prof. Klinkers AR research group as a Ph.D. student in 2004. He is now involved in research to develop novel AR-supported driver assistance systems for head-up displays in cars.