Parallel Multi-Hypothesis Algorithm for Criticality Estimation in Traffic and Collision Avoidance

Eduardo Sanchez, Michael Botsch, Andreas Gaull
Ingolstadt Univ. of Applied Sciences

Alexander Kammenhuber, Christoph Lauer, Tobias Dirndorfer
Audi AG

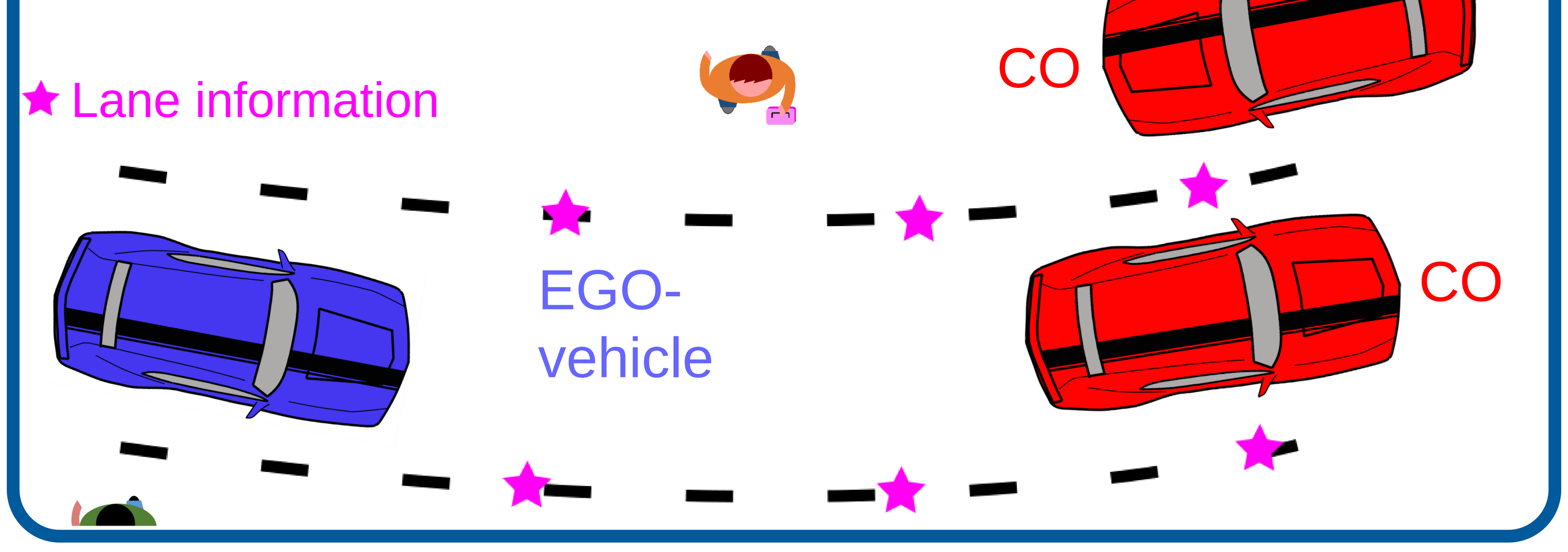
Richard Membarth, Philipp Slusallek
German Research Center for Artificial Intelligence (DFKI)

1. Overview

- Prediction of the collision probability of a traffic situation termed as “Criticality”
- Generation of traffic situations based on vehicle sensor information
- A model-based prediction of possible vehicle and pedestrian trajectories
- Detection of possible collisions between traffic participants
- Approximation of conditional probabilities of the possible collisions
- Implementation on multi-core processor
- Output of the algorithm: safe trajectories

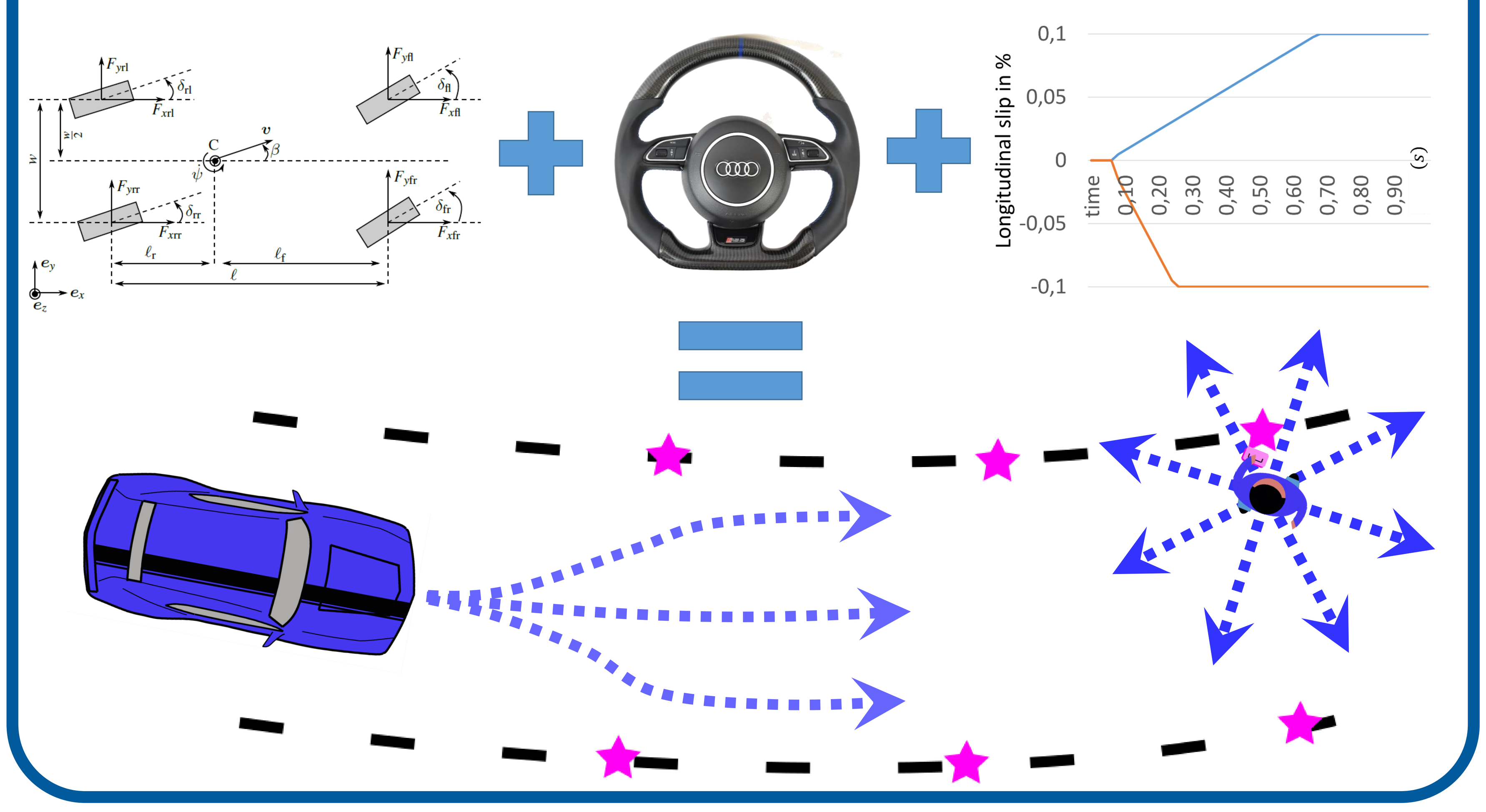
2. Traffic Situation Data Processing

- Sensors deliver information from the traffic situation:
  - Road infrastructure (lanes)
  - State of own (EGO-) vehicle
  - State of collision objects (CO)
    - Other vehicles
    - Pedestrians



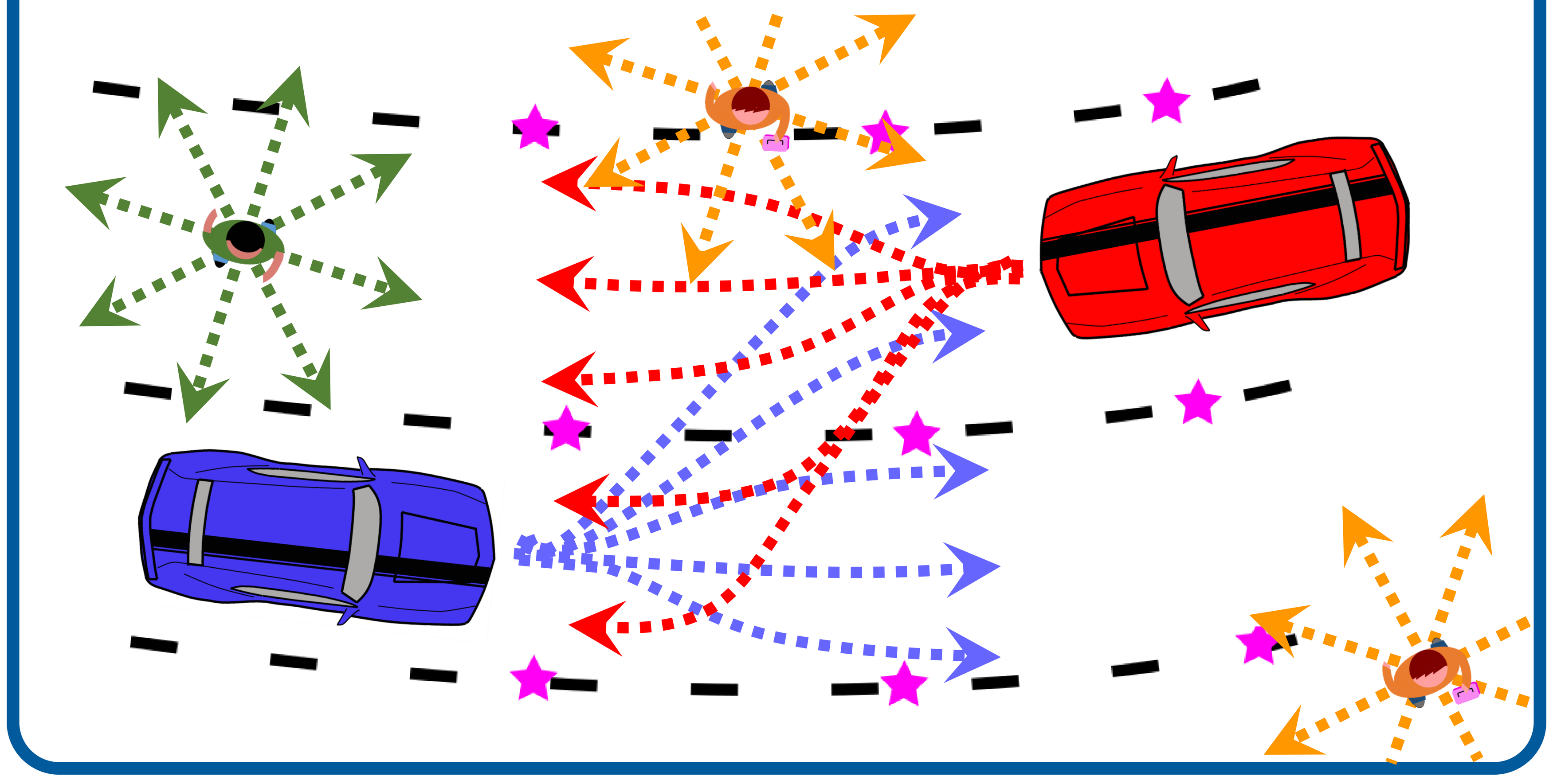
3. Trajectory Generation

- Generation of thousands of realistic trajectories for all vehicles by combining
  - Motion models
  - Controllers for lateral dynamics
  - Acceleration profiles
- Pedestrians are not bound to the road infrastructure



4. Collision Recognition

- Comparison between millions of pose combinations (EGO+CO)
- Detection of possible collisions between objects



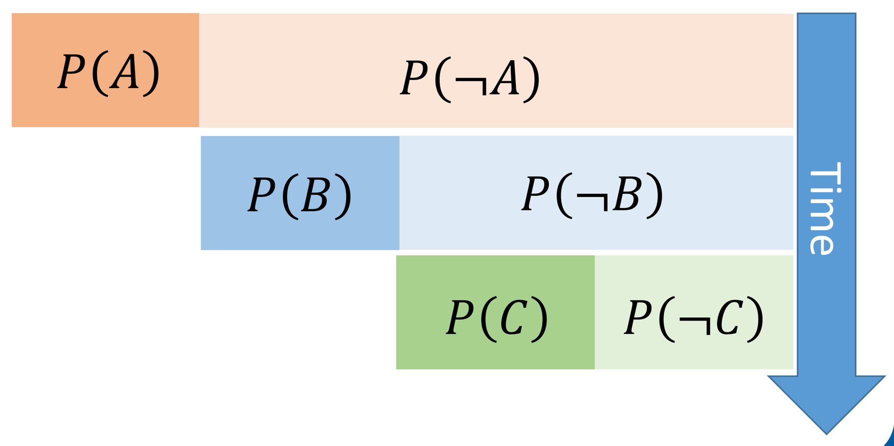
5. Risk Assessment

- Considering two objects**:** joint probability of two objects driving trajectories that lead to a collision
- Extension to more objects**:** taking into account the order in which the collisions can occur

$P(A)$

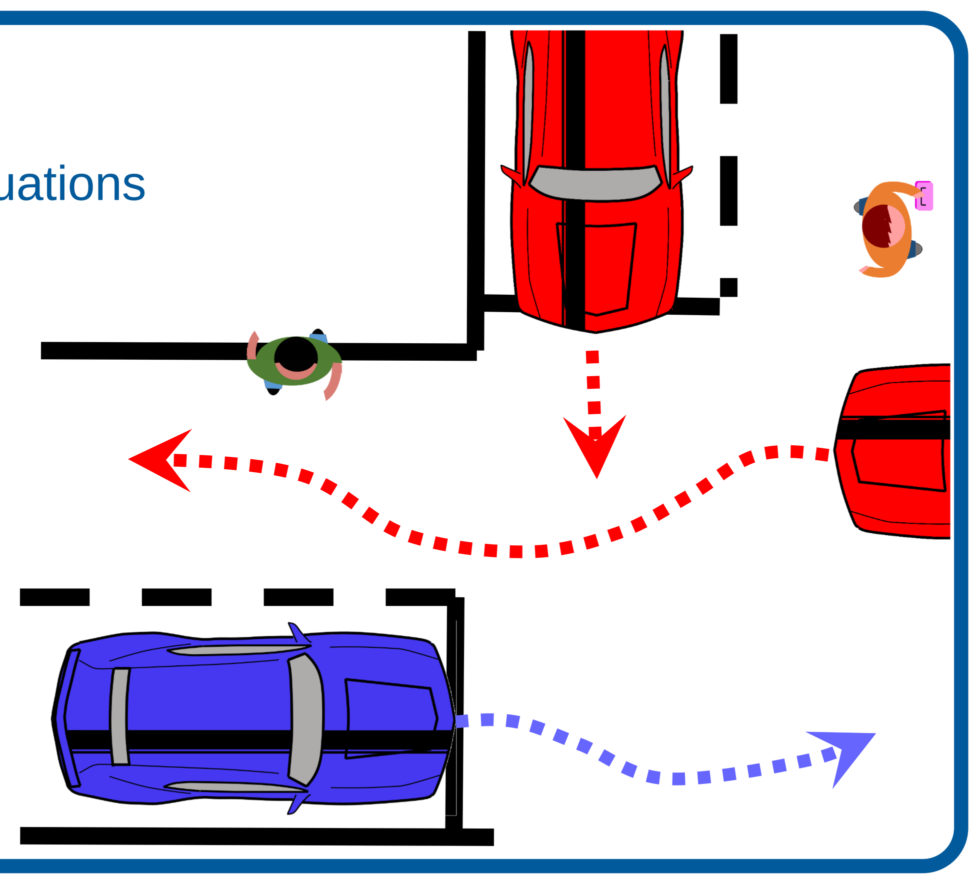
$P(B \cap \neg A) \approx P(\neg A) \cdot P(B)$

$P(C \cap \neg B \cap \neg A) \approx P(\neg A) \cdot P(\neg B) \cdot P(C)$



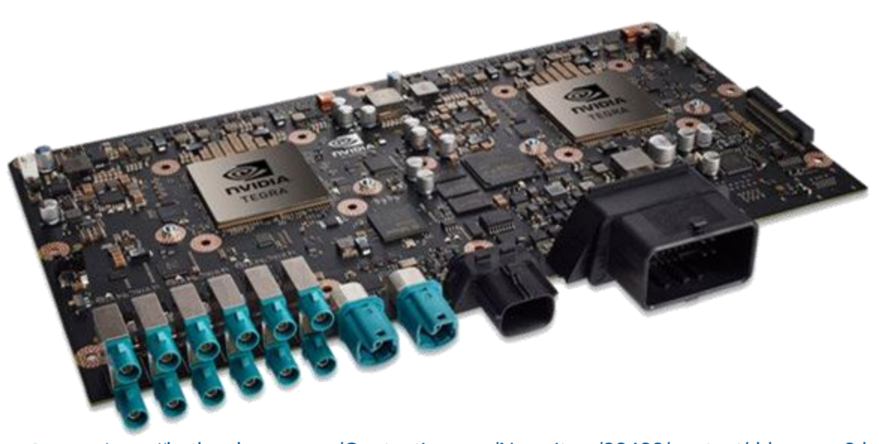
6. Evaluation

- 20 generic, designed traffic situations
- 547 real-life traffic situations



7. Algorithm Mapping to Embedded Systems

- Possibility of parallel execution of thousands of threads
- Exploit hardware specific features: module mapping to the GPU, extensive use of intrinsics (sine, cosine, tangent), no data-transfers between CPU and GPU
- AnyDSL used to avoid tying the implementation to a specific platform
- Thrift used to retrieve input data from the vehicle and to return the algorithm outputs



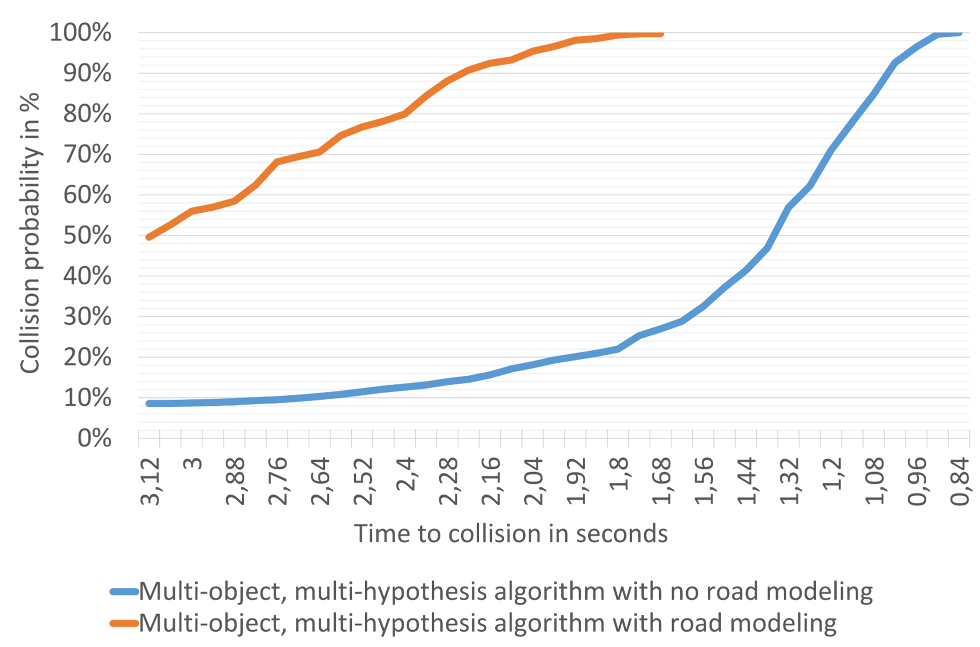
Source: https://hothardware.com/ContentImages/NewsItem/38428/content/drive_px_2.jpg

Runtime performance on a Drive PX 2

# of CO	# of pose combinations	CPU ARM Cortex A57	Pascal GPU GP106 - 1280 cores	Tegra X2 GPU GP10B - 256 cores
3	25.93 million	1800 ms	10 ms	15 ms
	18.00 million	1600 ms	8 ms	11 ms
10	86.44 million	11600 ms	21 ms	49 ms
	60.01 million	9600 ms	14 ms	35 ms

8. Results

- Smooth, gradual changes in the predicted criticality
- Enough anticipation time for activating passive safety systems (cross-traffic) and active safety systems (longitudinal traffic)



9. Conclusion

- Prediction of safe trajectories
- A fully model-based, multi-modal, parallelizable algorithm is presented.
- The criticality of the current traffic situation is predicted
- Further inclusion of road infrastructure elements is possible
- Implementation on vehicle-compatible hardware proves prototypically possibility of use in series-production vehicles