2024 FIE-SCSE Doctoral Seminar

Artificial Intelligence Techniques in Autonomous Driving

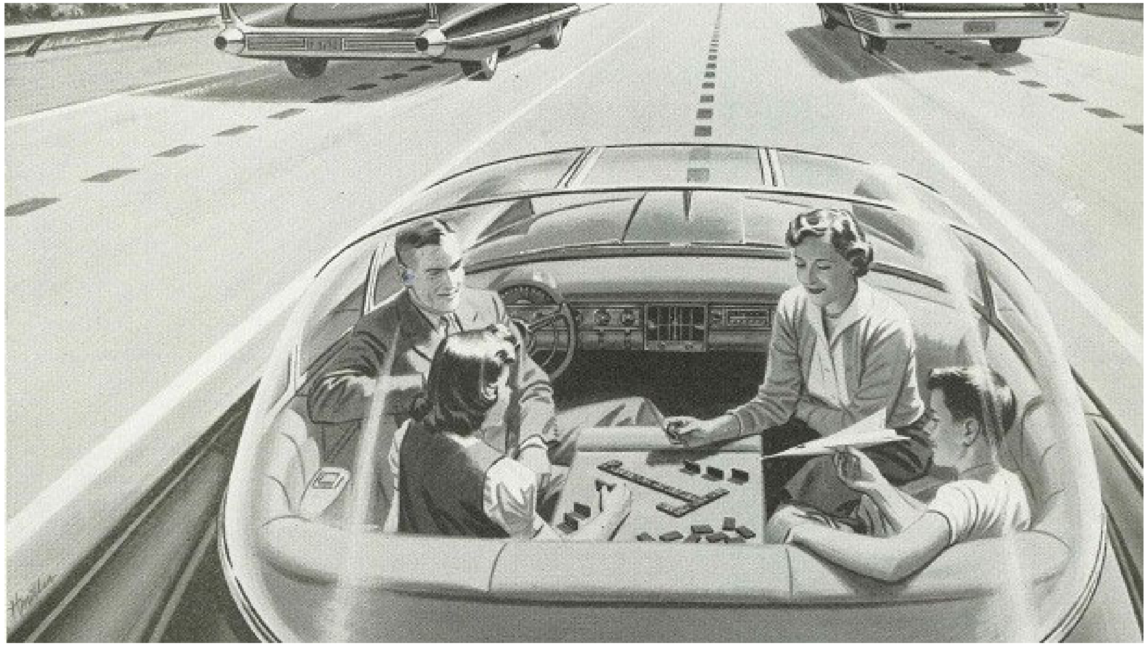
Dagang Li

School of Computer Science and Engineering

Content

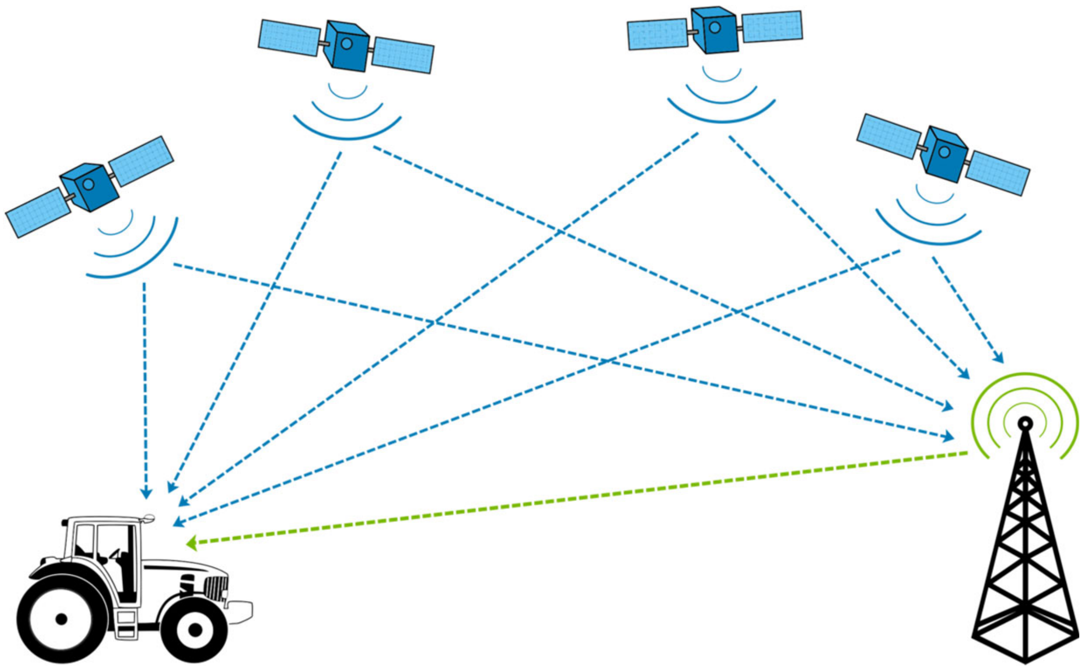
Autonomous Driving / Self-driving / Automated Driving

- Human Dream: relaxed journey
- No-driver freight transportation
- Eliminate human errors



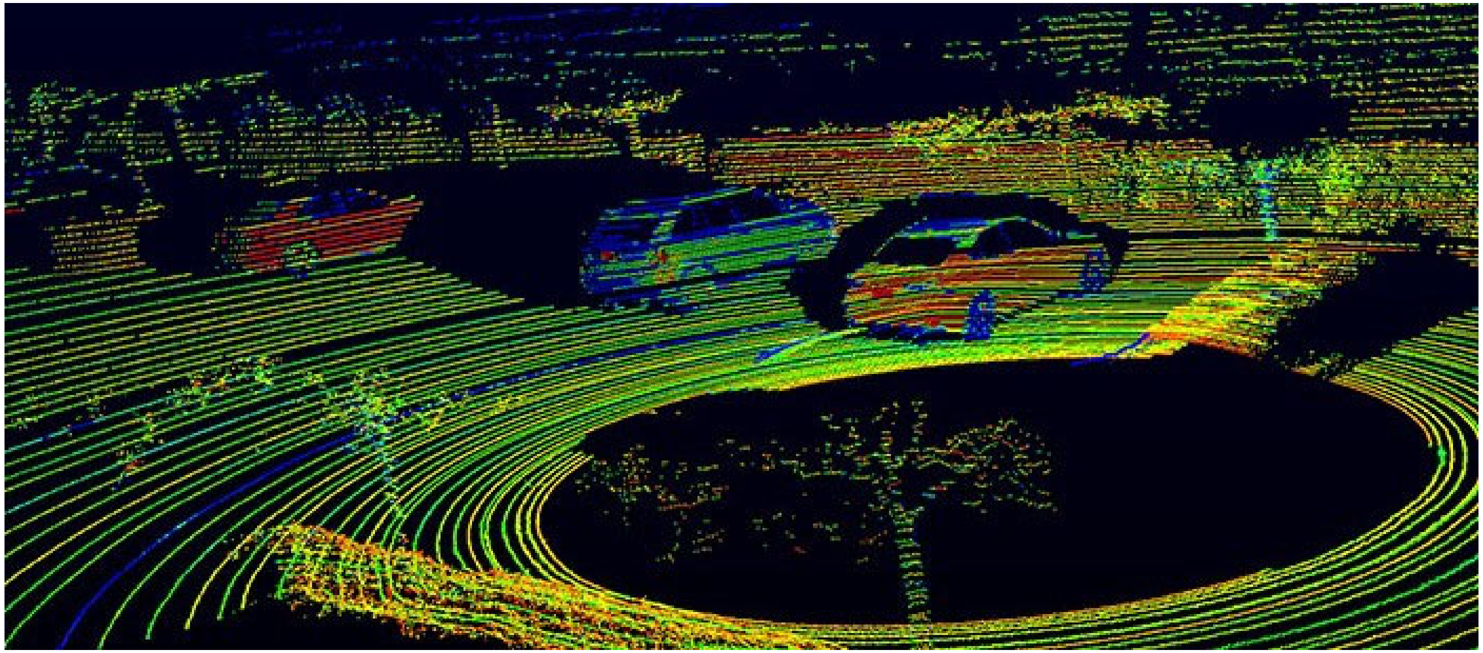
- Mobility for elderly and people with disabilities
- New ways of public transportation
- Reduce number of cars

2000: First Technological Revolution: GPS, IMUs & Maps



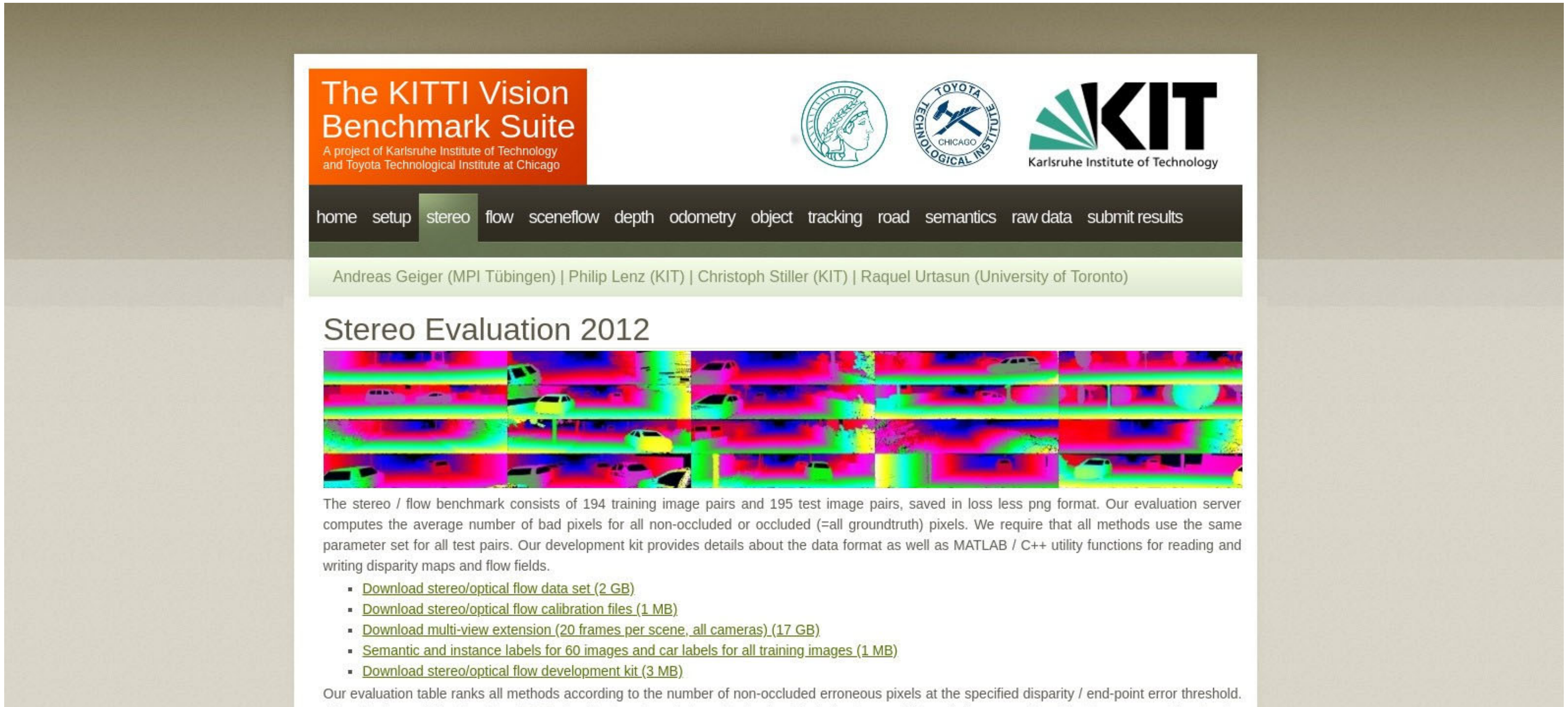
- NAVSTAR GPS available with 1 meter accuracy, IMUs improve up to 5 cm
- Navigation systems and road maps available
- Accurate self-localization and ego-motion estimation algorithms

2006: Second Technological Revolution: Lidars & High-res Sensors



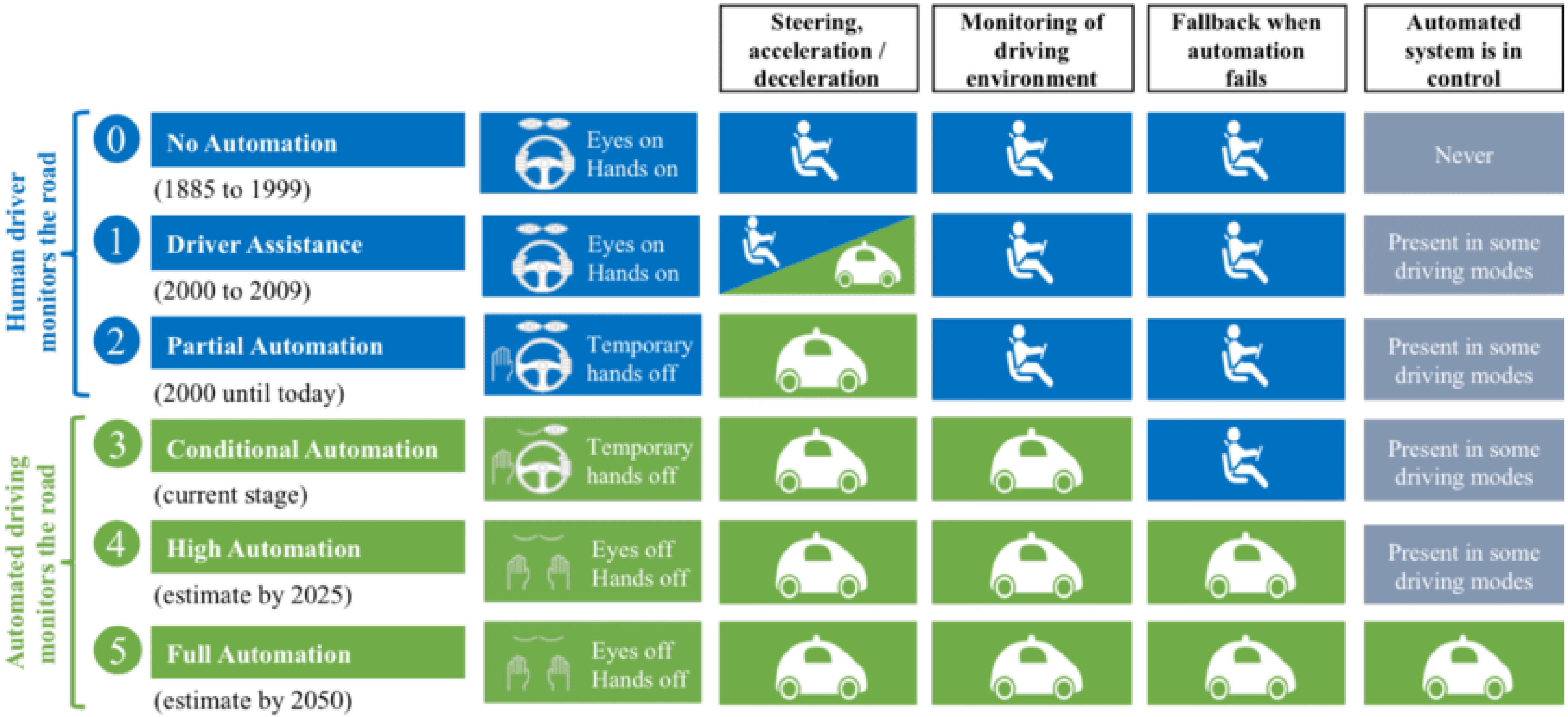
- High-resolution Lidar
- Camera systems with increasing resolution
- Accurate 3D reconstruction, 3D detection & 3D localization

2012: Third Technological Revolution: Deep Learning



- Representation learning boosts in accuracy across tasks and benchmarks

2014: Society of Automotive Engineers: SAE Levels of Autonomy

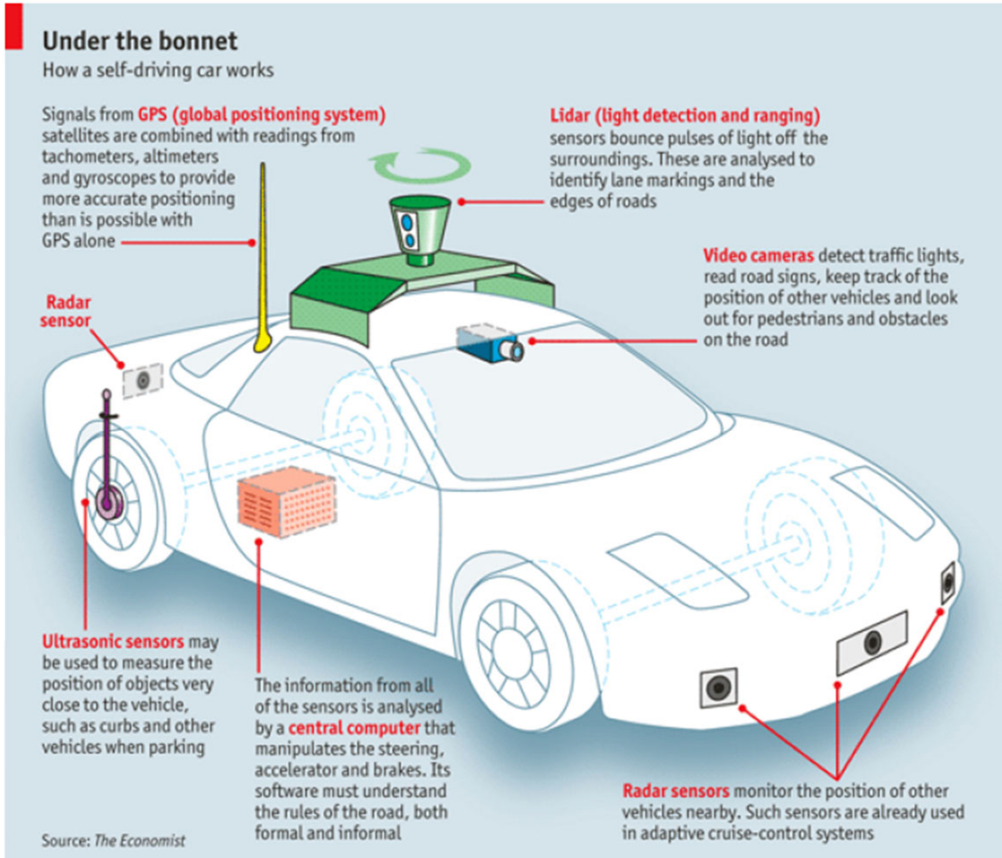


Evolution to Autonomous Driving

- Autonomous or nothing (Google, ~~Apple~~, Uber, Huawei)
  - Very risky, only few companies can do this
  - Long term goals

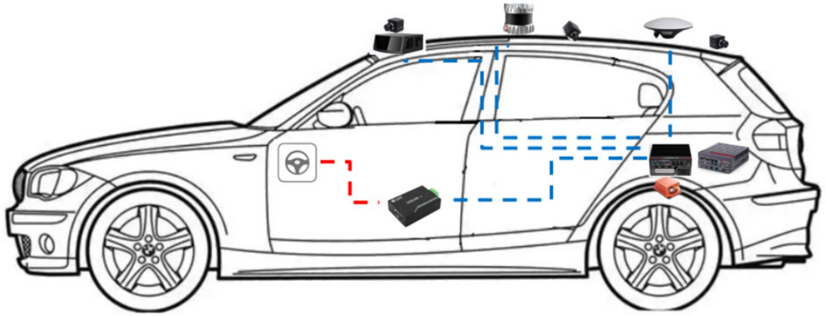
- Introduce technology little by little (all car companies)
  - Car industry is very conservative
  - ADAS as intermediate goal
  - Sharp transition: how to maintain the driver engaged?

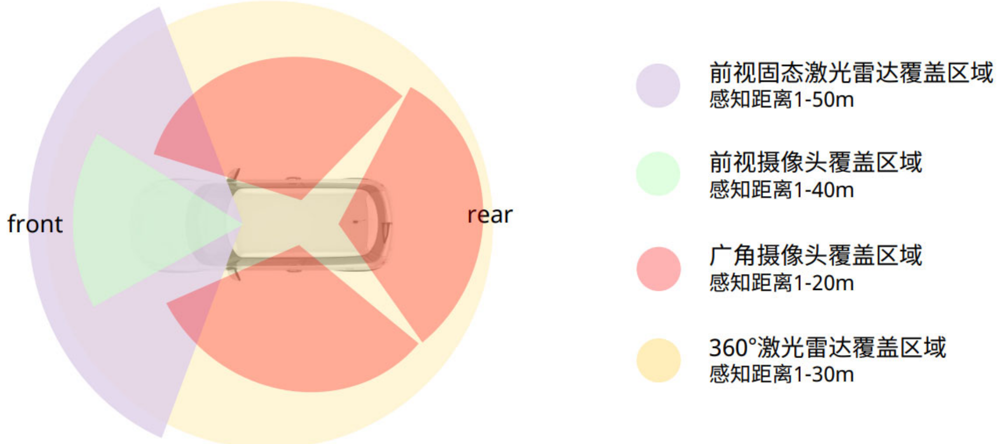
Basic Physical Ecosystem of an Autonomous Vehicle



- Global Positioning System (GPS)
- Light Detection and Ranging (LIDAR)
- Cameras (Video)
- Ultrasonic Sensors
- Central Computer
- Radar Sensors
- Dedicated Short-Range Communications-Based Receiver (not pictured)

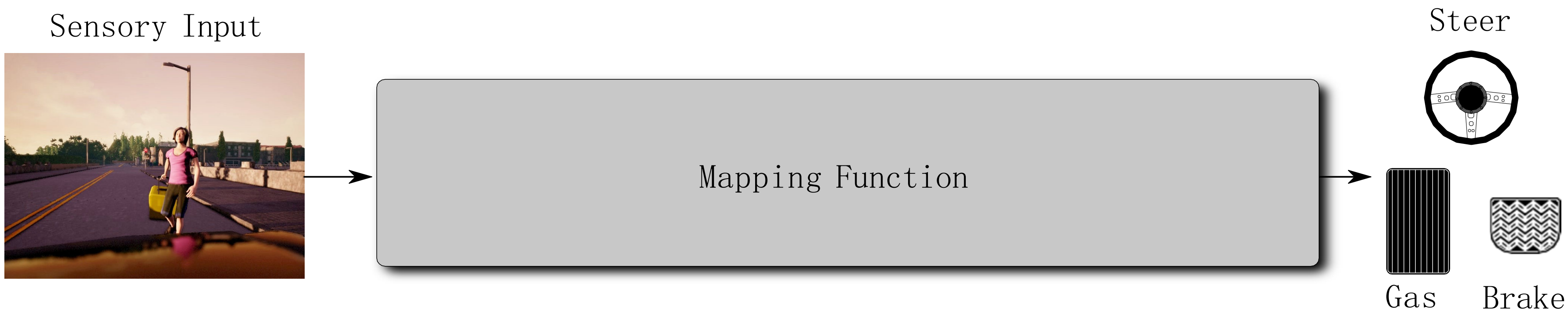
MUST Open Intelligent Electric Vehicles Research Center (OIEV-RC)





Autonomous Driving Approaches

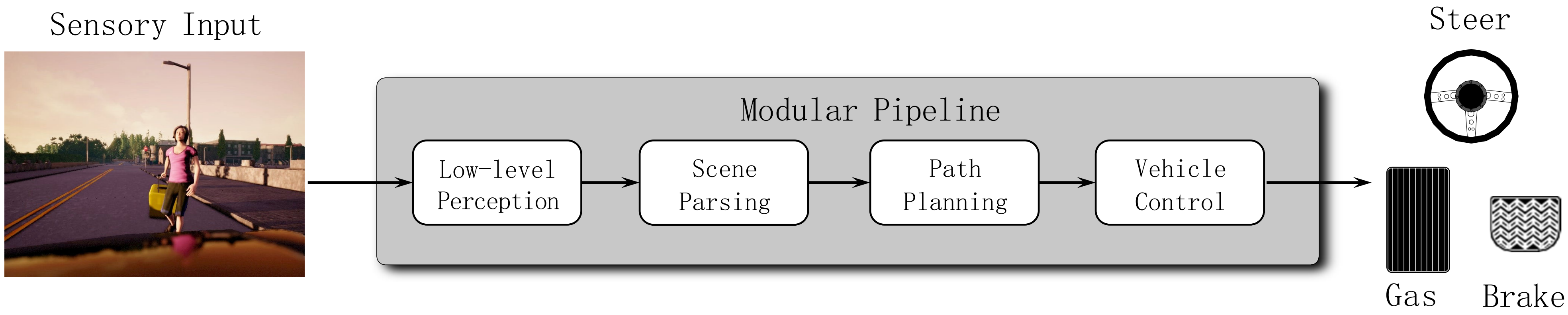
Autonomous Driving



Dominating Paradigms:

- Modular Pipelines
- End-to-End Learning (Imitation Learning, Reinforcement Learning)
- Direct Perception

Autonomous Driving: Modular Pipeline



Pros:

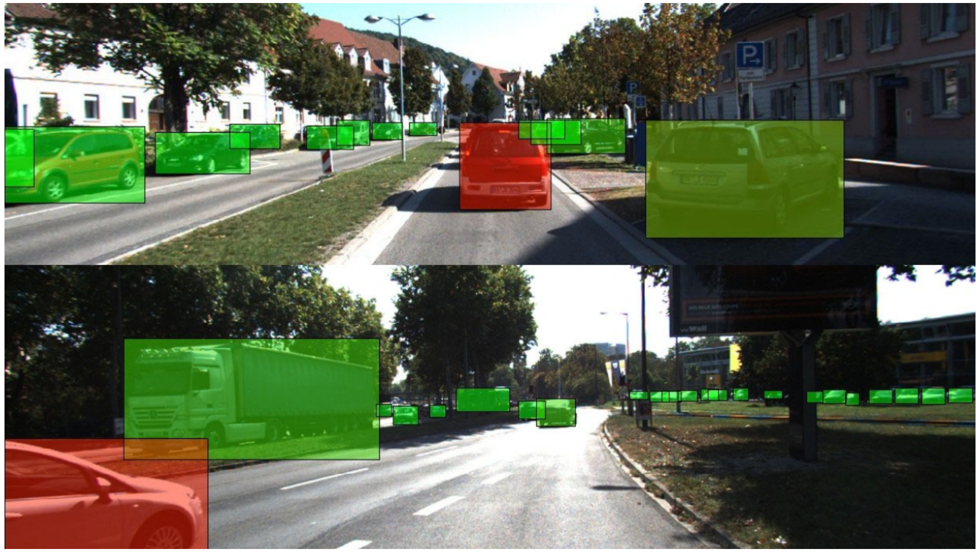
- Small components, easy to develop in parallel
- Interpretability

Cons:

- Piece-wise training (not jointly)
- Localization and planning heavily relies on HD maps

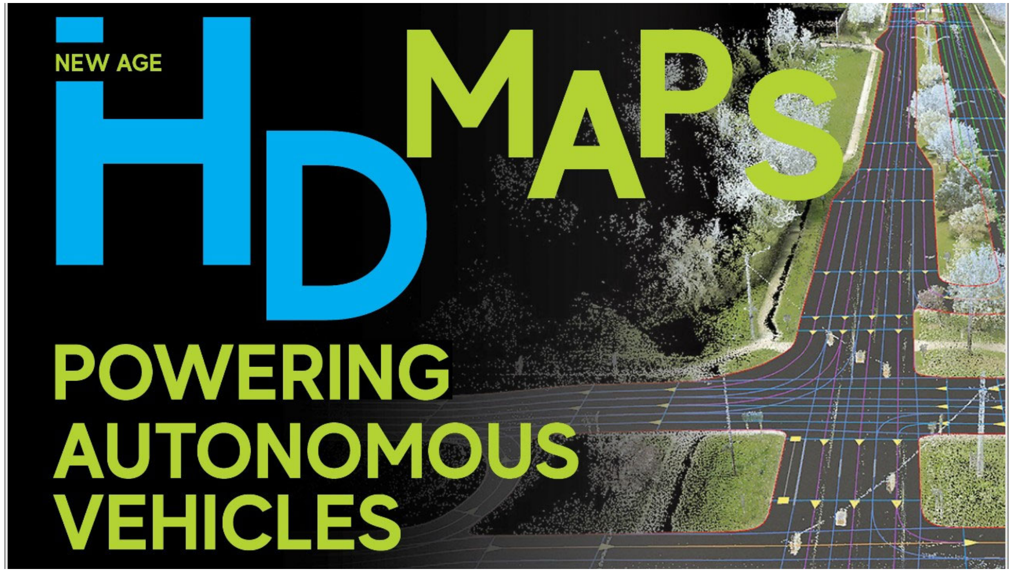
HD maps: Centimeter precision lanes, markings, traffic lights/signs, human annotated

Autonomous Driving: Modular Pipeline

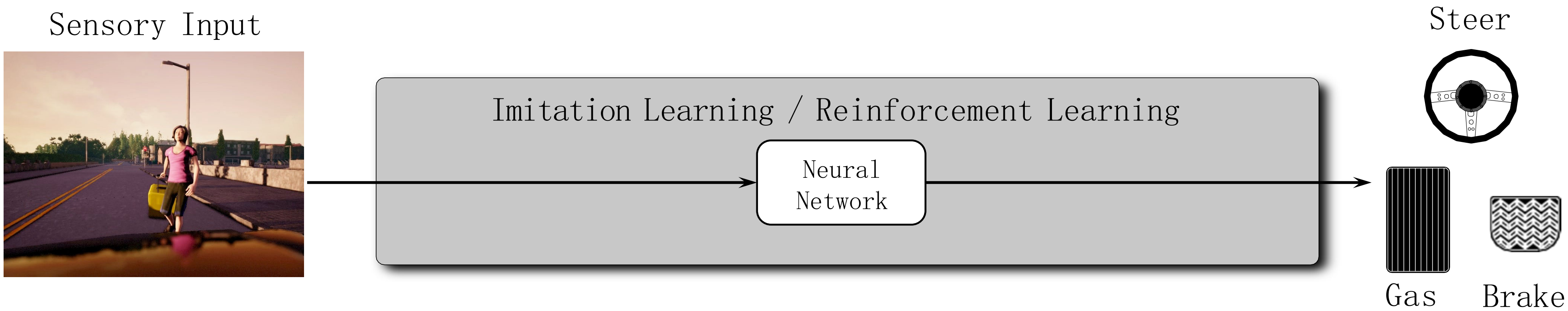


Piece-wise training difficult: not all objects are equally important!

HD Maps are expensive to create (data collection & annotation effort)



Autonomous Driving: End-to-End Learning



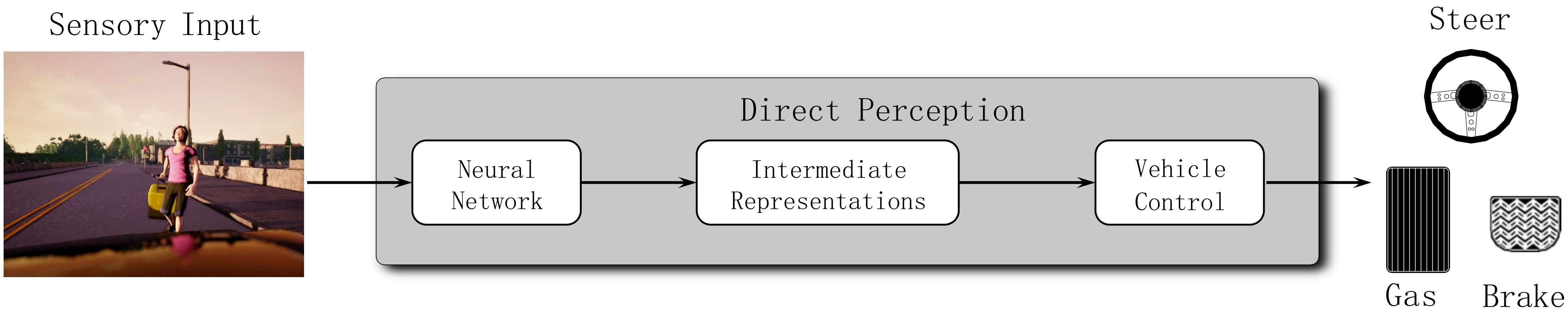
Pros:

- End-to-end training
- Simple
- Cheap annotations

Cons:

- Training / Generalization
- Interpretability
- Data hungry

Autonomous Driving: Direct Perception



Pros:

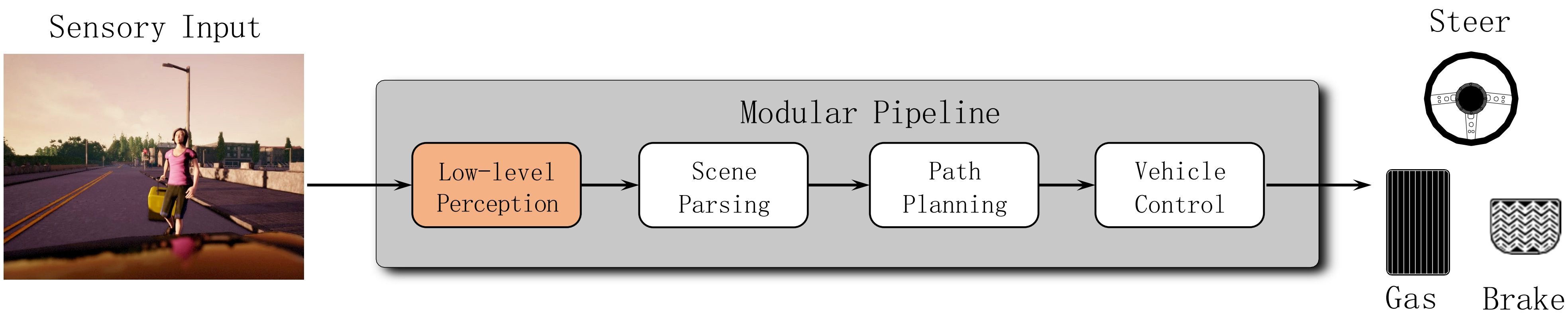
- Compact Representation
- Interpretability

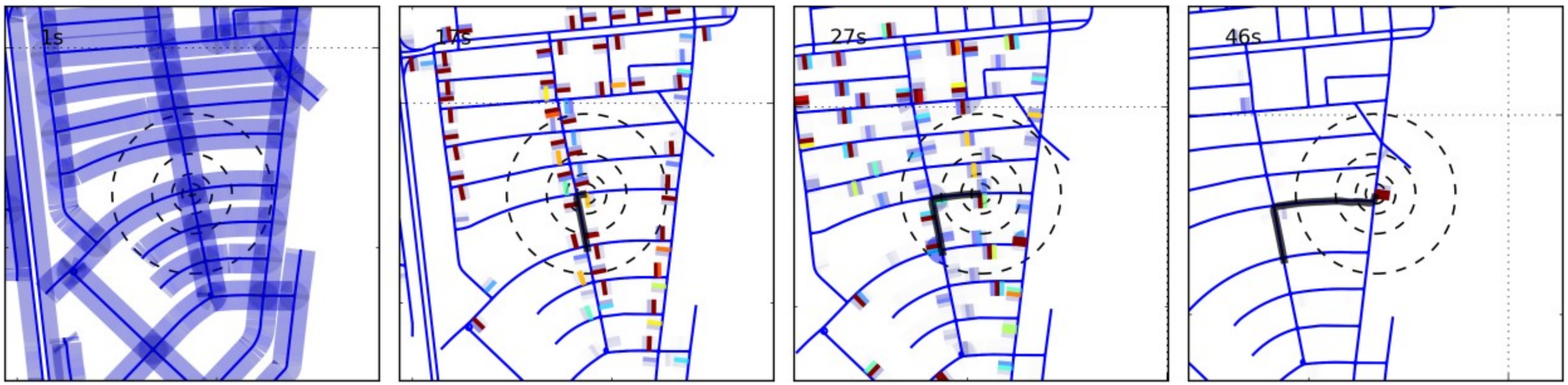
Cons:

- Control typically not learned jointly
- How to choose representations?

Autonomous Driving: Modular Pipeline

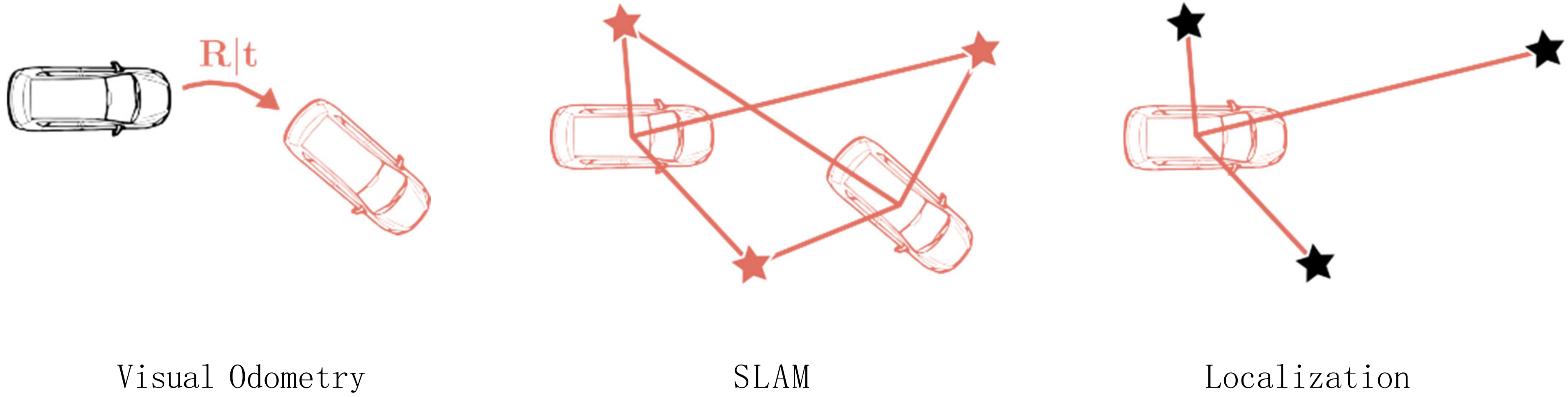
Autonomous Driving: Modular Pipeline





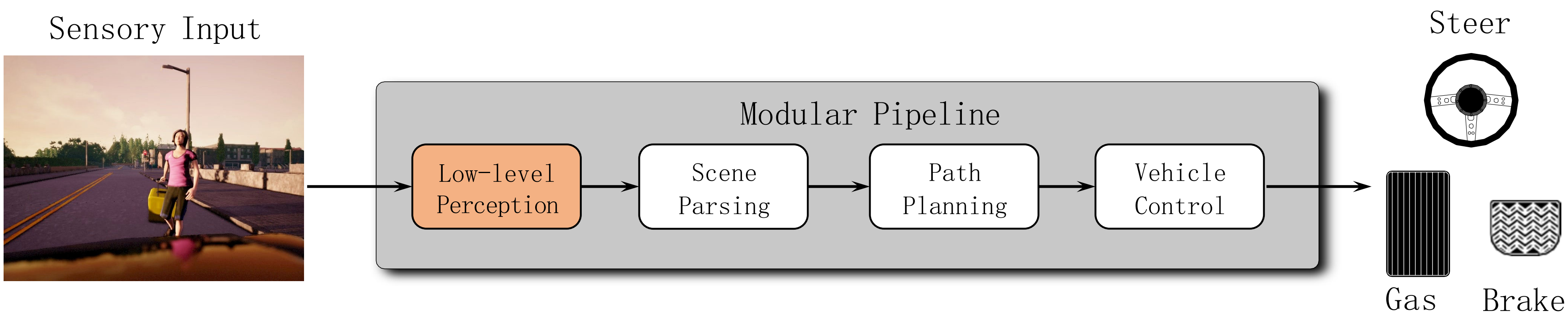
Ego-motion Estimation and Localization

Self-driving cars operate in a dynamic environments and hence must be aware of their ego-motion (own motion) and the motion of other traffic participants.

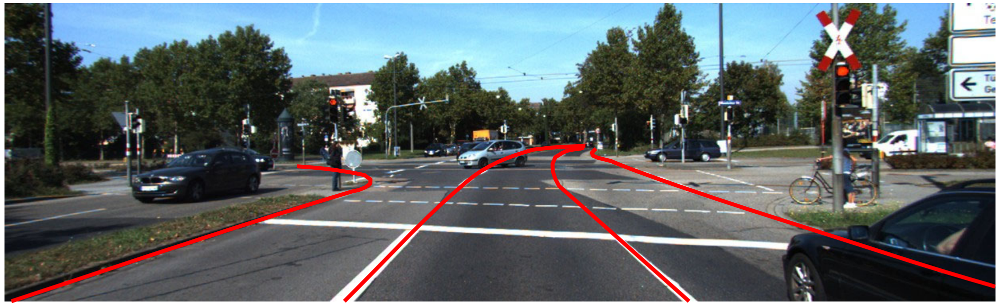


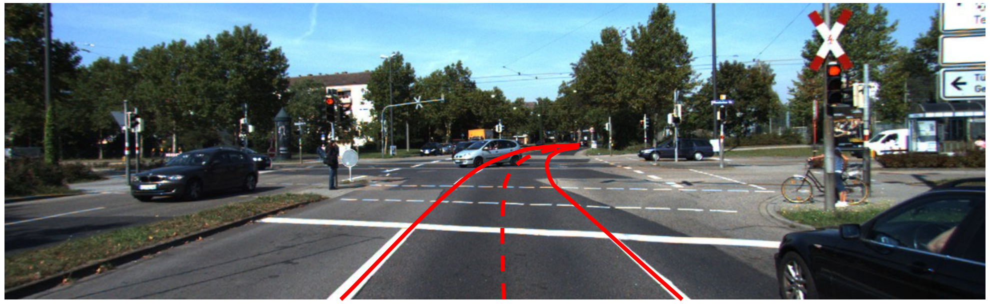
- **Visual odometry** refers to the estimation of relative ego-motion from images
- **SLAM algorithms** build a map and simultaneously localize in that map
- **Localization methods** find the global pose of a vehicle in a given map

Autonomous Driving: Modular Pipeline



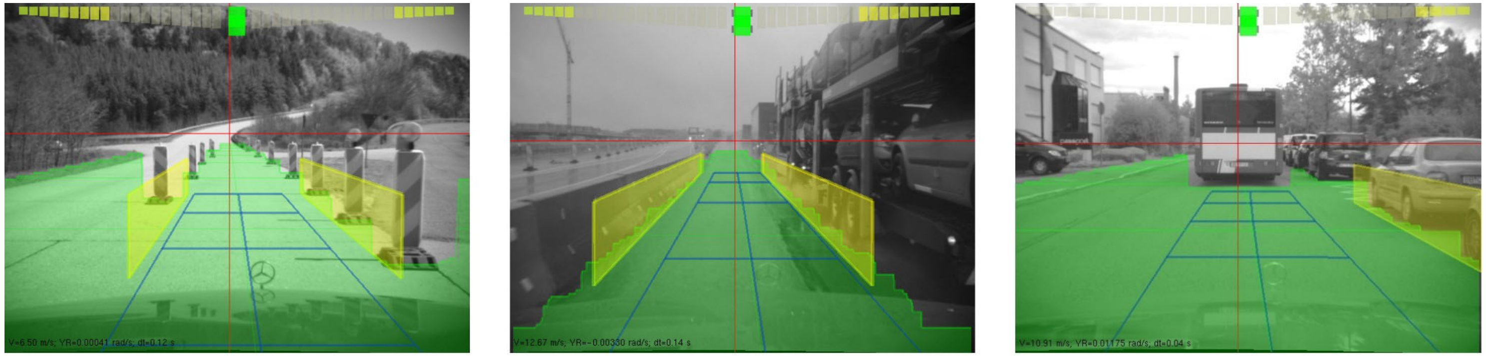
Road and Lane Detection



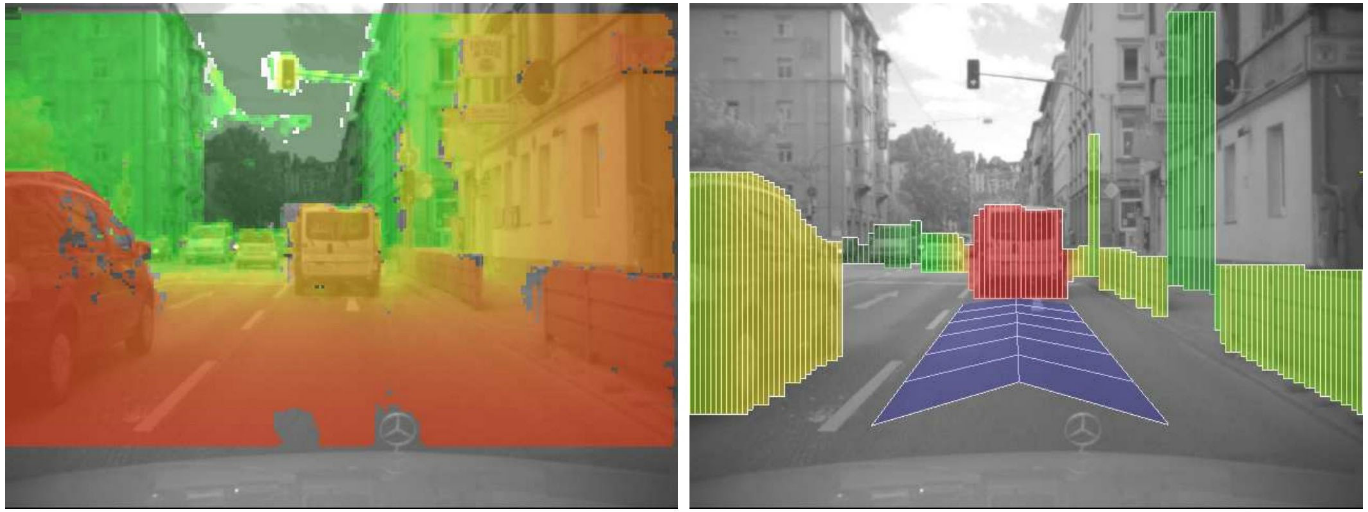


Occupancy Map

Height of obstacles unknown / only freespace estimation



(a) Highway. (b) Freeway. (c) Downtown.

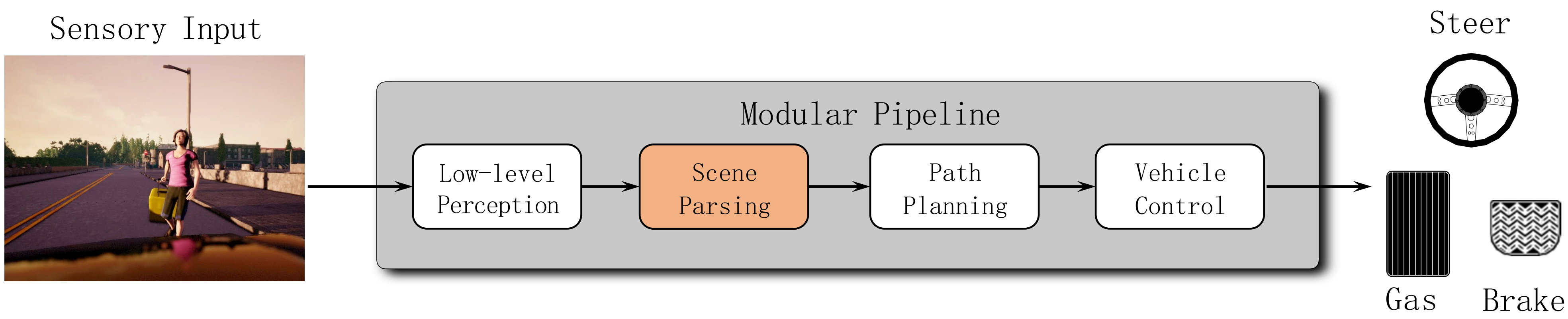


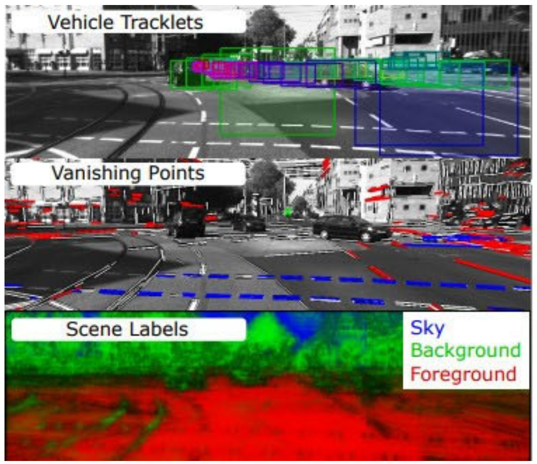
(a) Dense disparity image (SGM result) (b) Stixel representation

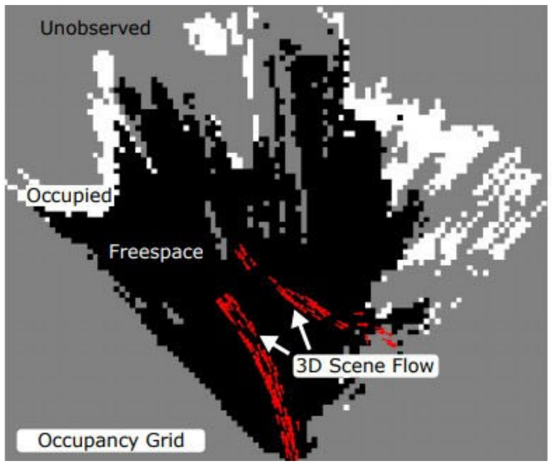
The Stixel World

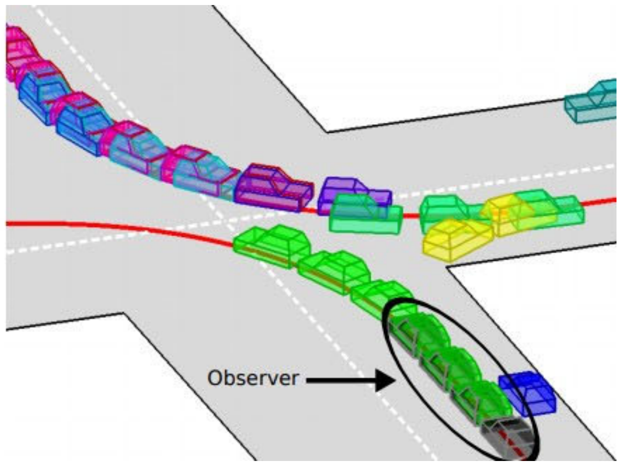
“Stixel World” estimates multiple stixels per image column (i.e., segmentation)

Autonomous Driving: Modular Pipeline

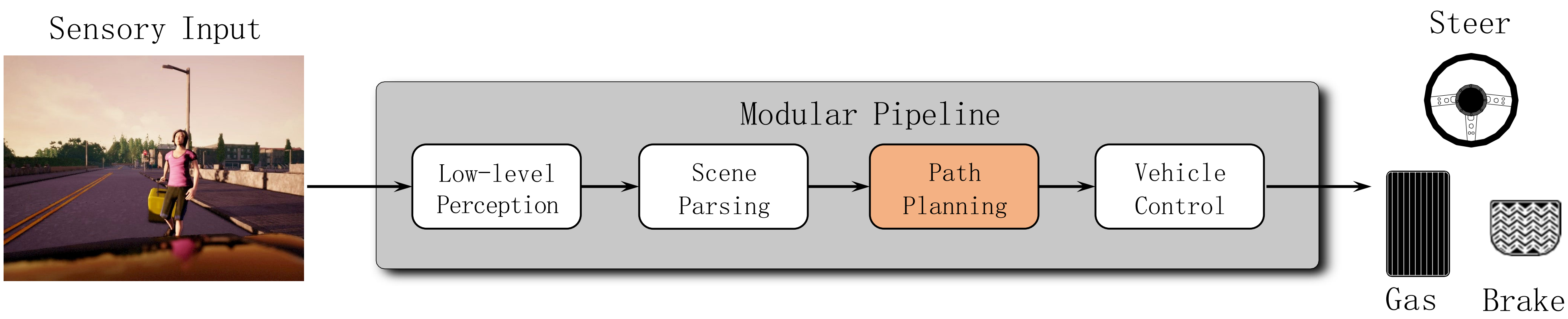


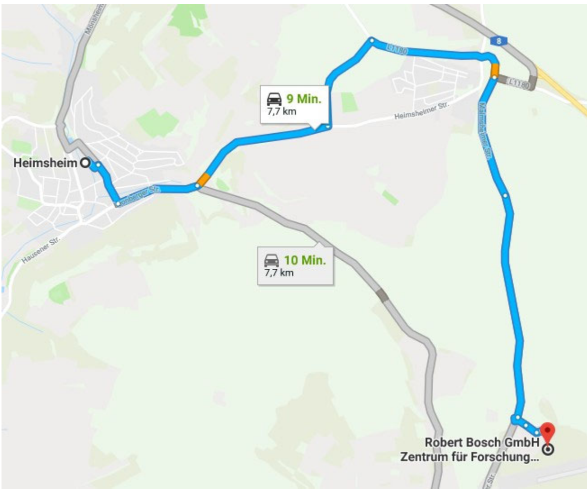


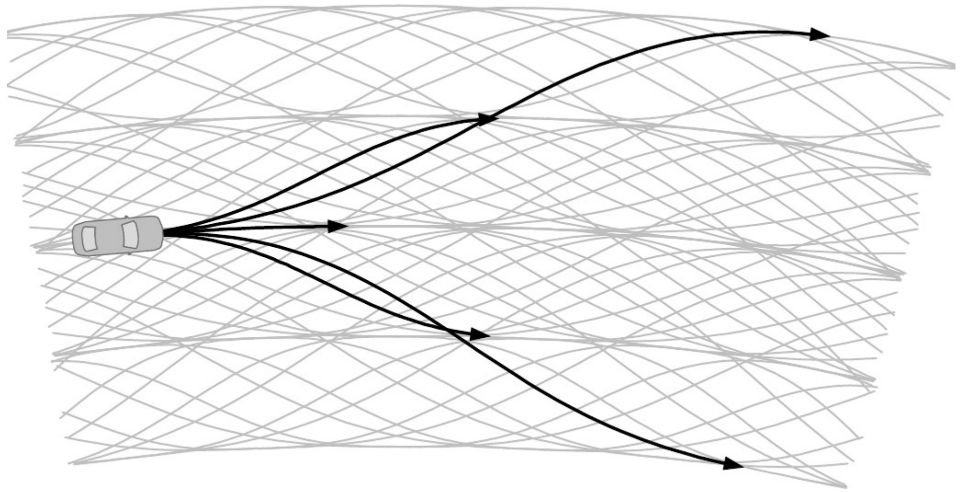




Autonomous Driving: Modular Pipeline



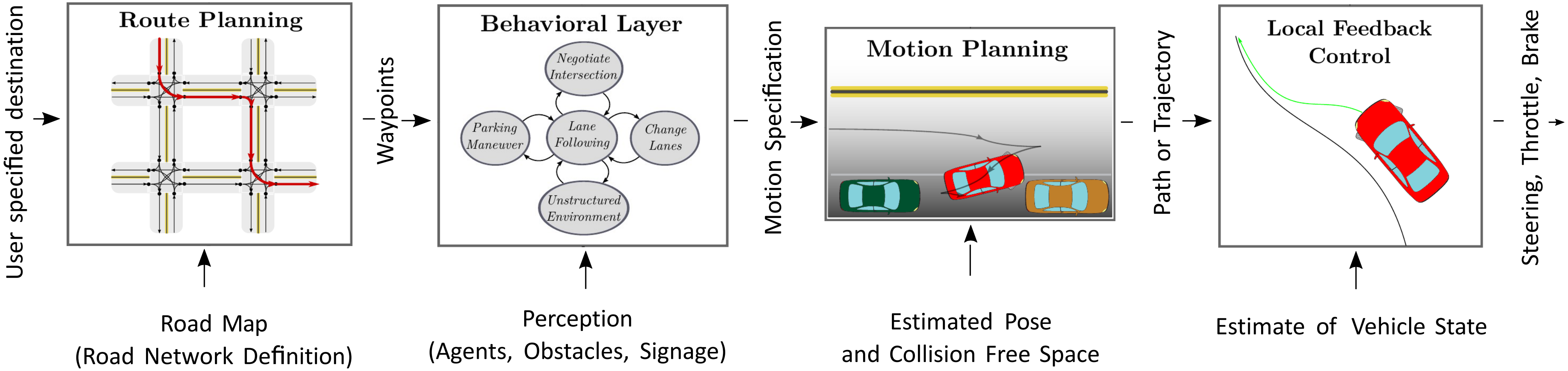




Planning and Decision Making

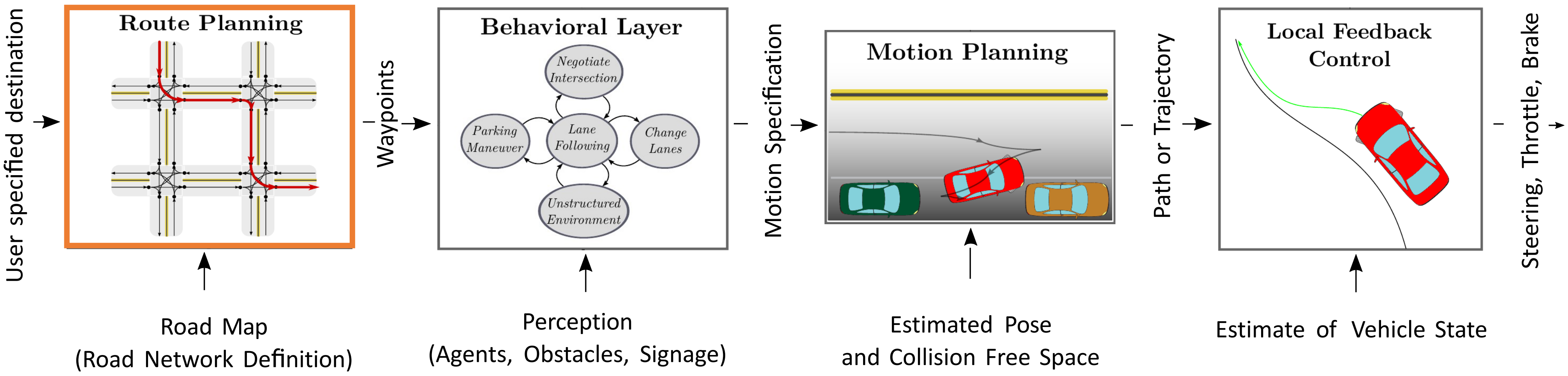
- Problem Definition:
  - Goal: find and follow path from current location to destination
  - Take static infrastructure and dynamic objects into account
  - Input: vehicle and environment state (via perception stack)
  - Output: path or trajectory as input to vehicle controller
- Challenges:
  - Driving situations and behaviors are very complex
  - Thus difficult to model as a single optimization problem

Planning and Decision Making



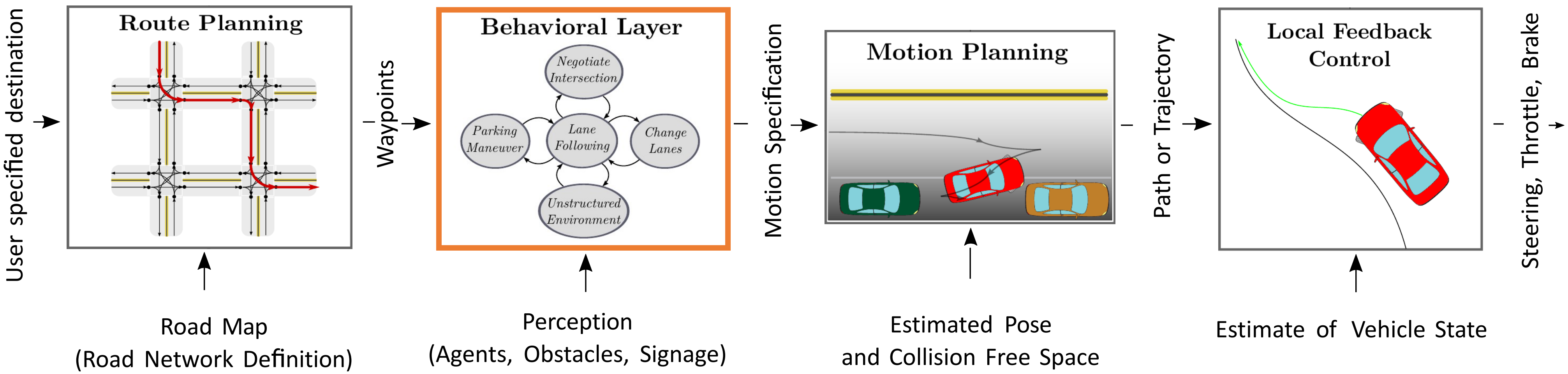
- Idea: Break planning problem into a **hierarchy of simpler problems**
- Each problem tailored to its scope and level of abstraction
- Earlier in this hierarchy means higher level of abstraction
- Each optimization problem will have constraints and objective functions

Step 1: Route Planning



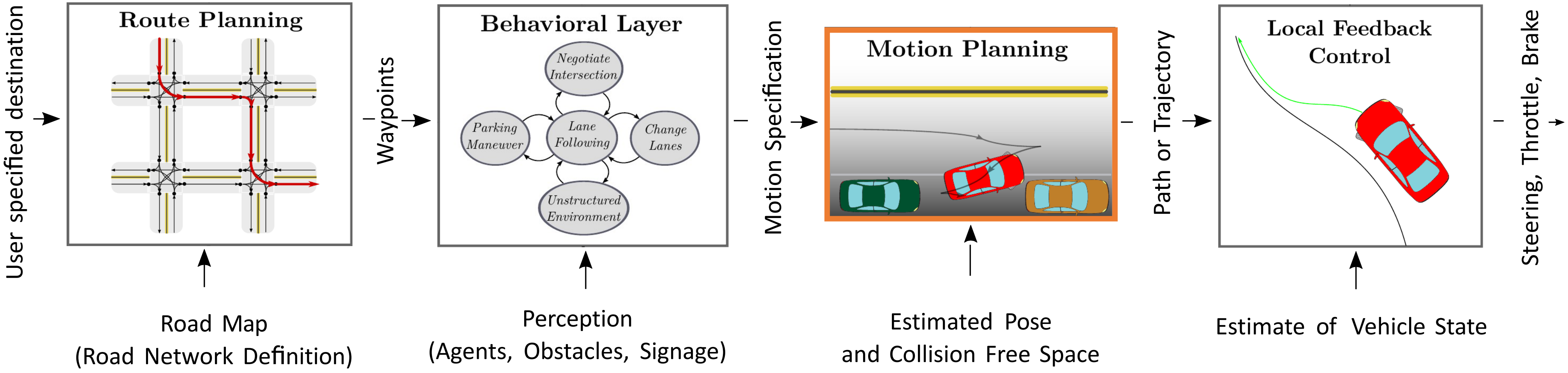
- Represent road network as directed graph
- Edge weights correspond to road segment length or travel time
- Problem translates into a minimum-cost graph network problem
- Inference algorithms: Dijkstra, A$*$, . . .

Step 2: Behavior Planning



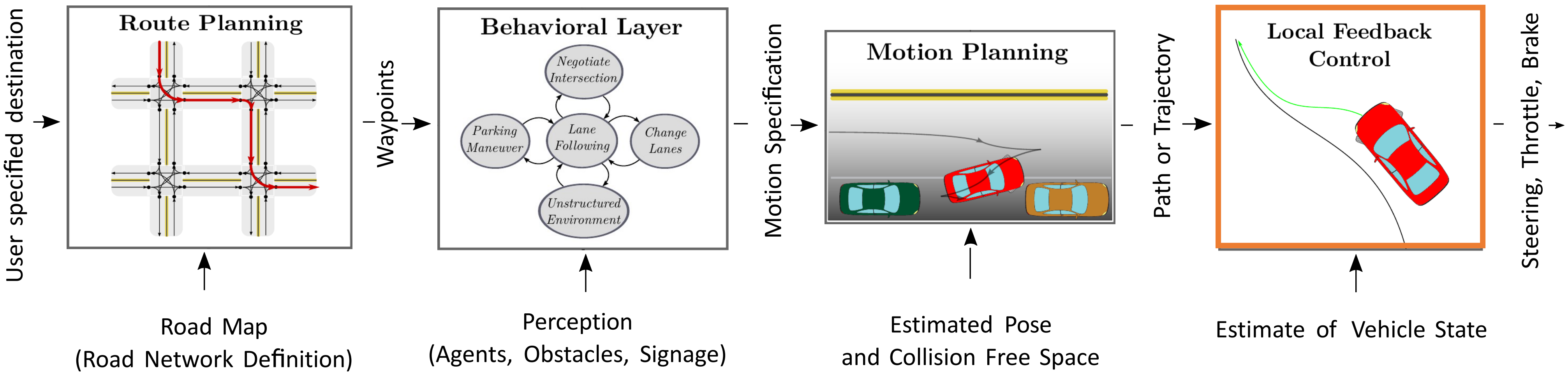
- Select driving behavior based on current vehicle/environment state
- E.g. at stop line: stop, observe other traffic participants, traverse
- Often modeled via finite state machines (transitions governed by perception)
- Can be modeled probabilistically, e.g., using Markov Decision Processes (MDPs)

Step 3: Motion Planning



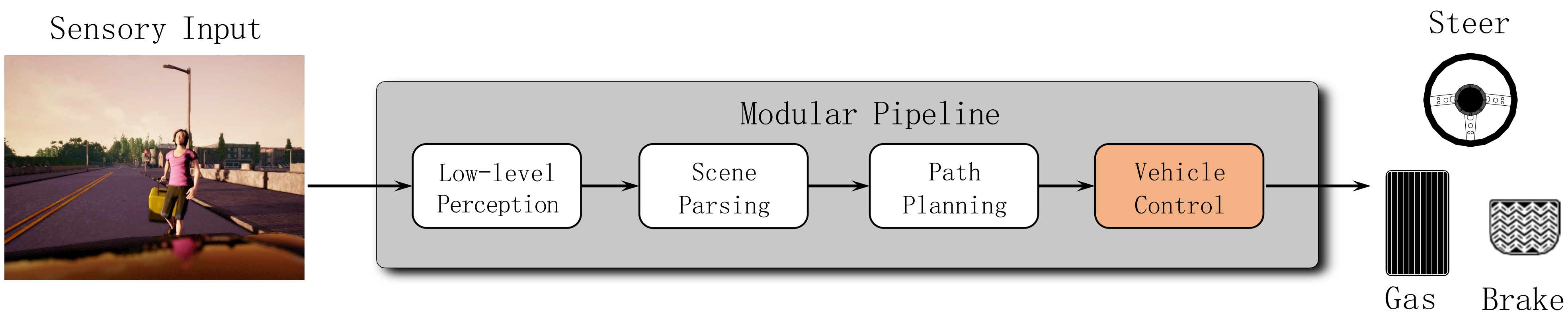
- Find feasible, comfortable, safe and fast vehicle path/trajectory
- Exact solutions in most cases computationally intractable
- Thus often numerical approximations are used
- Approaches: variational methods, graph search, incremental tree-based

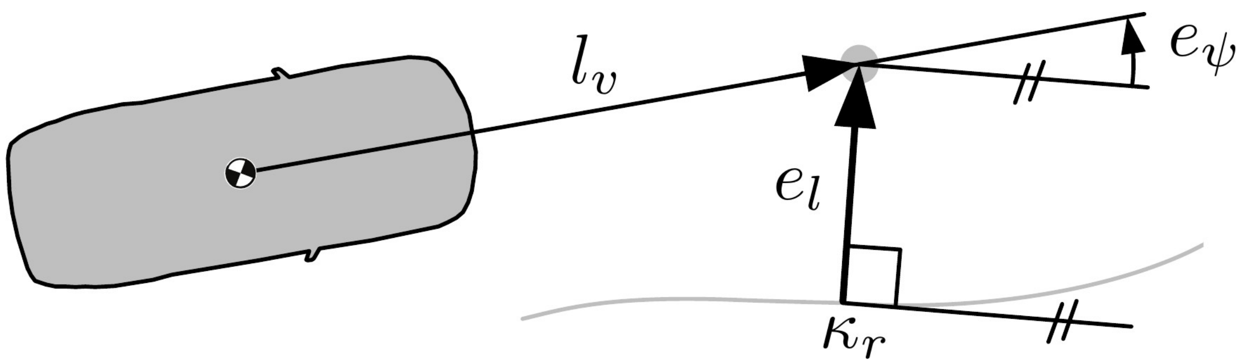
Step 4: Local Feedback Control



- Feedback controller executes the path/trajectory from the motion planner
- Corrects errors due to inaccuracies of the vehicle model
- Emphasis on robustness, stability and comfort
- Interact with underlying vehicle dynamics and control

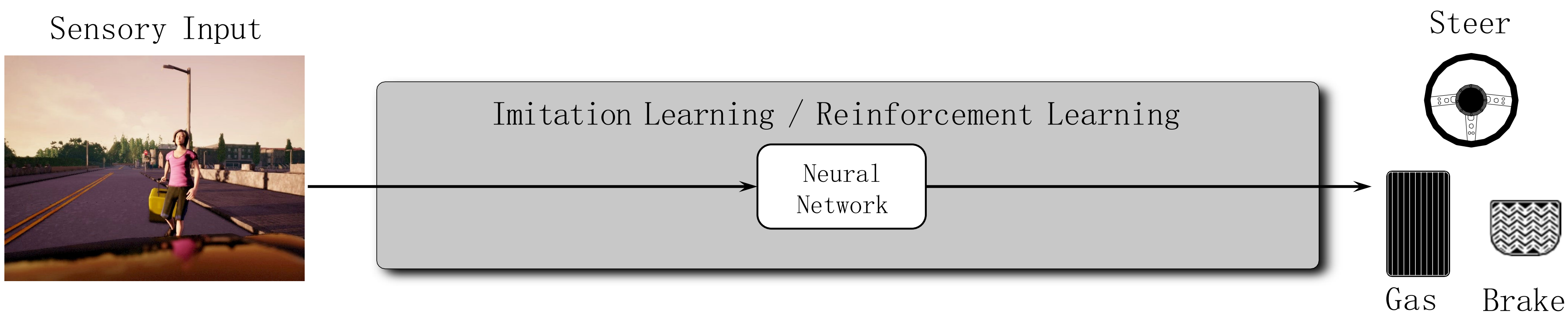
Autonomous Driving: Modular Pipeline





Autonomous Driving: End-to-end Learning

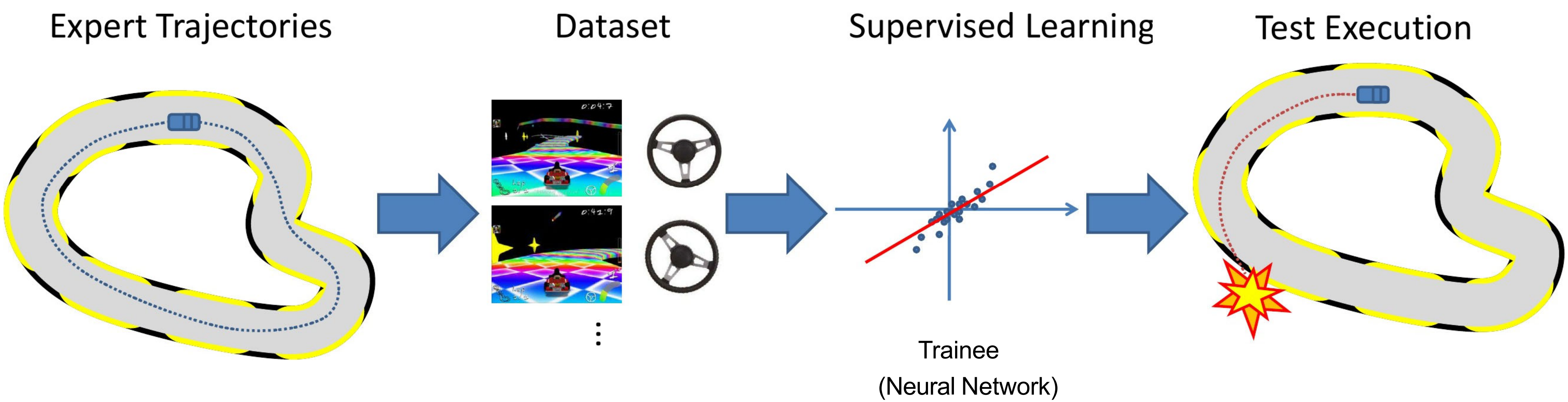
Autonomous Driving: End-to-End Learning



Idea:

- Generates ego-motion directly from the sensory input
- Perception, prediction and planning can be jointly trained and optimized towards the ultimate task
- Shared backbones increase computational efficiency
- Allow continuous learning to sense and act

Imitation Learning in a Nutshell



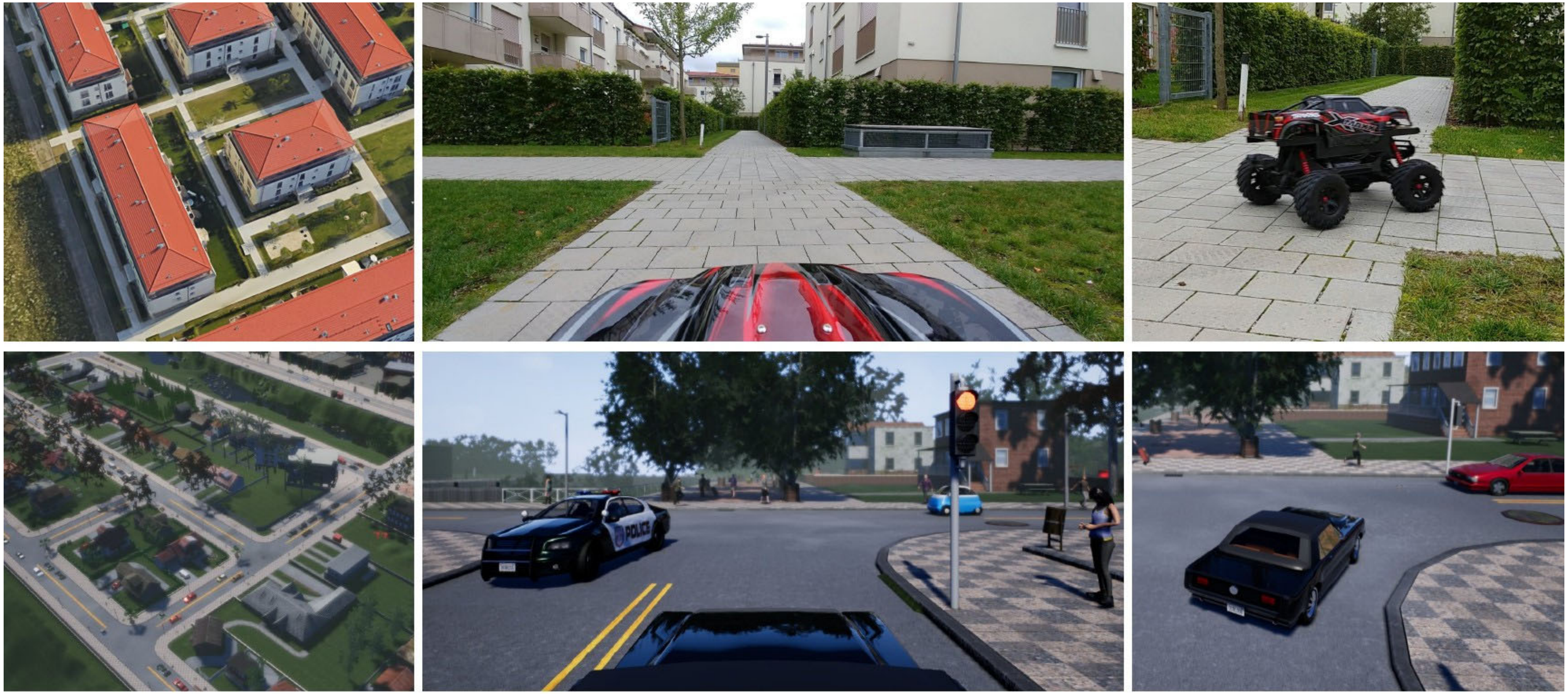
Hard coding policies is often difficult ⇒ Rather use a data-driven approach!

- **Given**: demonstrations or demonstrator
- **Goal**: train a policy to mimic decision
- **Variants**: behavior cloning (this lecture), inverse optimal control, . . .

Challenges of Behavior Cloning

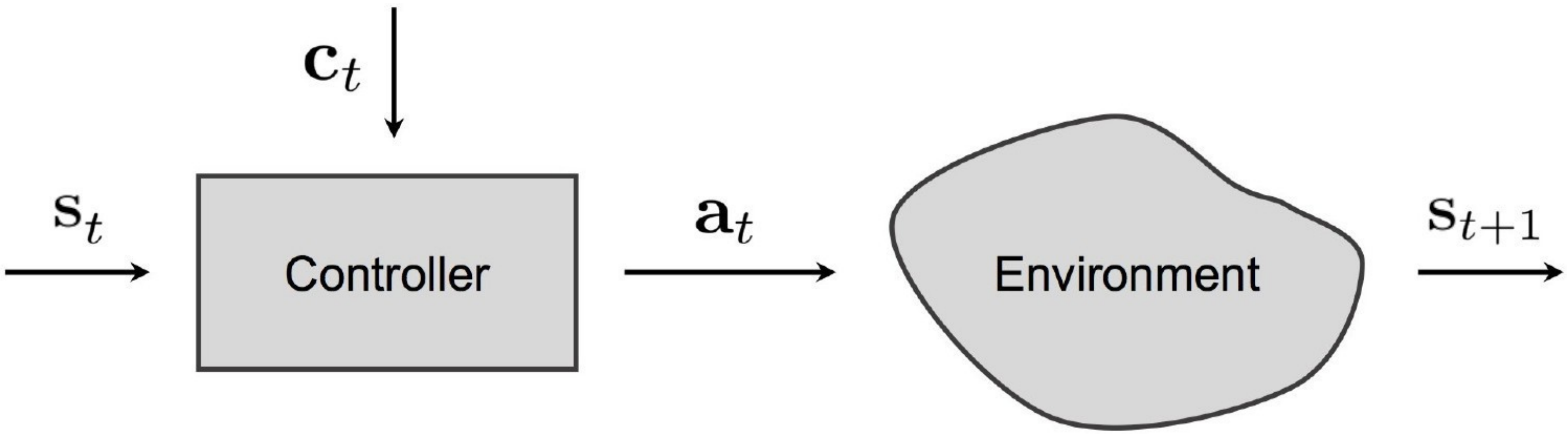
- Behavior cloning makes IID assumption
  - Next state is sampled from states observed during expert demonstration
  - Thus, next state is sampled independently from action predicted by current policy
- What if the trainee network makes a mistake?
  - Enters new states that haven't been observed before
  - New states not sampled from same (expert) distribution anymore
  - Cannot recover, catastrophic failure in the worst case
- What can we do to overcome this train/test distribution mismatch?

Conditional Imitation Learning



(a) Aerial view of test environment (b) Vision-based driving, view from onboard camera (c) Side view of vehicle

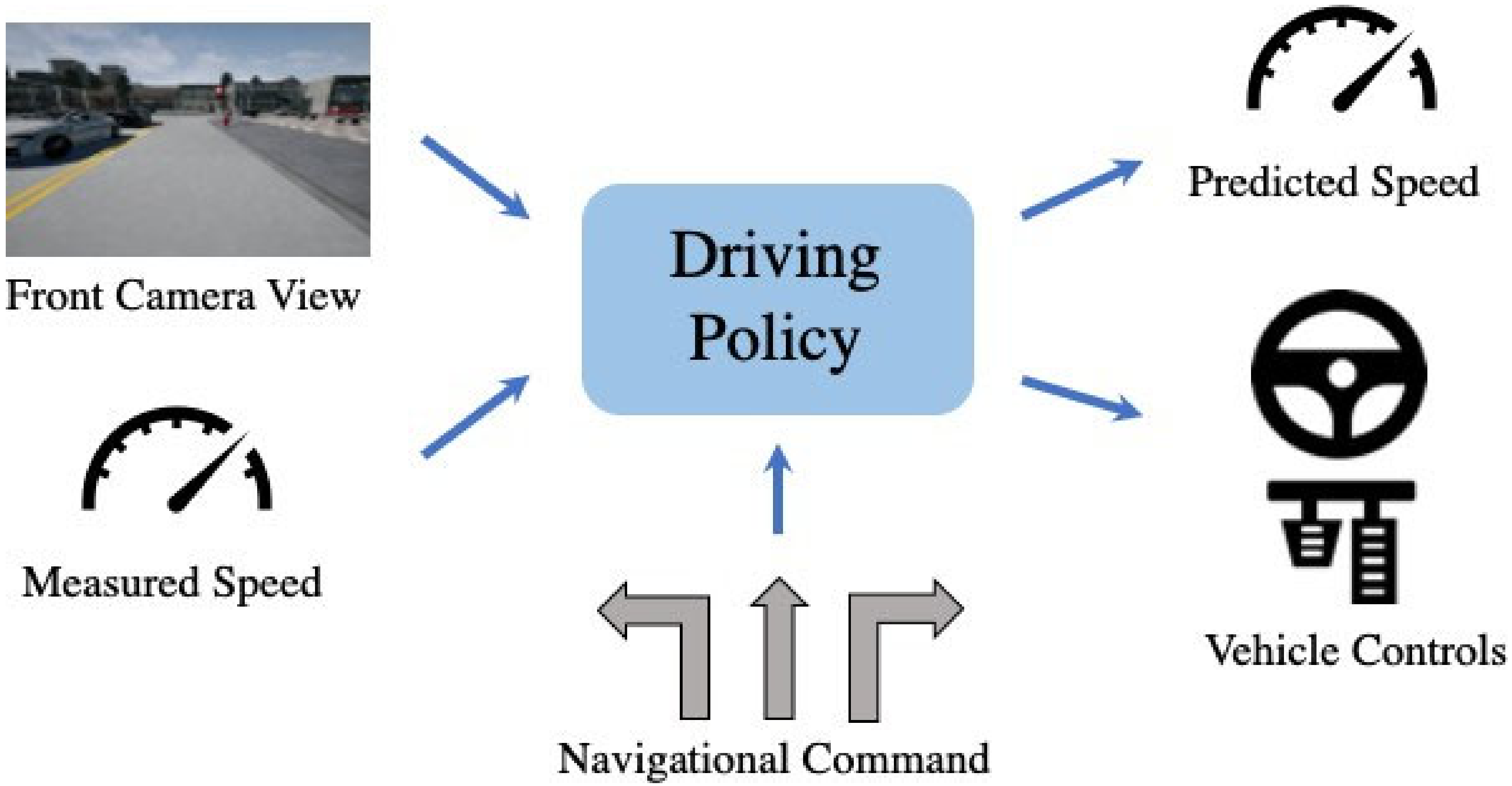
Conditional Imitation Learning



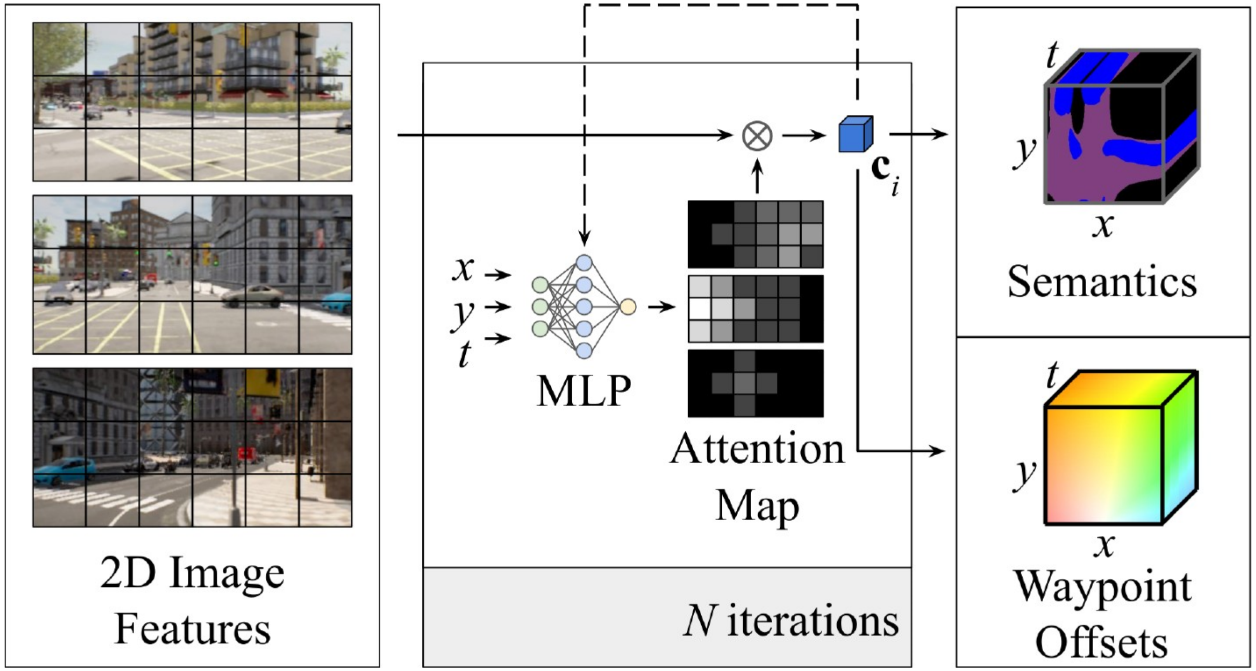
Idea:

- Condition controller on **navigation command** c ∈ {left, right, straight}
- High-level navigation command can be provided by consumer GPS, i.e., telling the vehicle to turn left/right or go straight at the next intersection
- This removes the task ambiguity induced by the environment
- State s_t: current image Action a_t: steering angle & acceleration

Conditional Imitation Learning



Neural Attention Fields



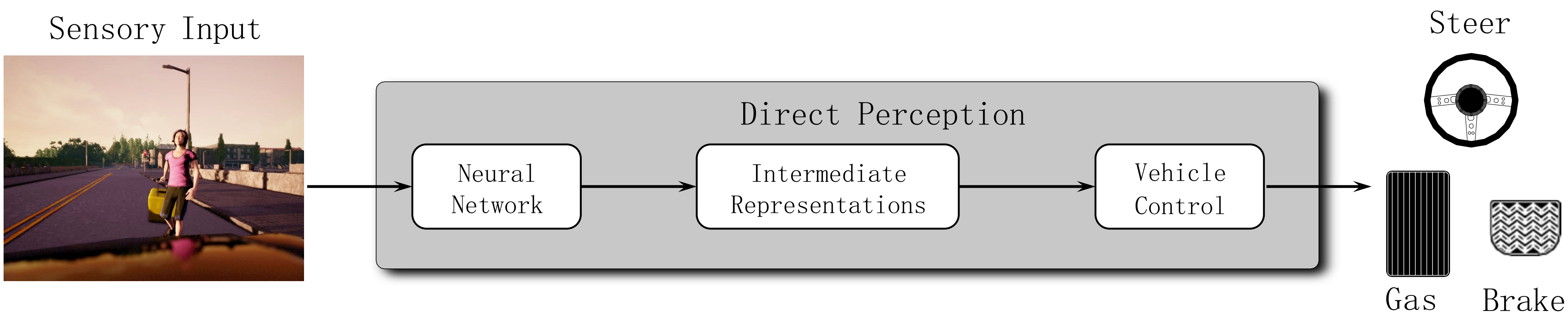
- An MLP iteratively **compresses** the high-dimensional input into a compact representation c_i based on a BEV query location as input
- The model predicts **waypoints** and auxiliary **semantics** which aids learning

Advantages and Challenges of Imitation Learning

- Advantages of Imitation Learning:
  - Easy to implement
  - Cheap annotations (just driving while recording images and actions)
  - Entire model trained end-to-end
  - Conditioning removes ambiguity at intersections
- Challenges for Imitation Learning?
  - Behavior cloning uses IID assumption which is violated in practice
  - Direct mapping from images to control $\Rightarrow$ No long term planning
  - No memory (can’t remember speed signs, etc.)
  - Mapping is difficult to interpret (“black box”), despite visualization techniques

Autonomous Driving: Direct Perception

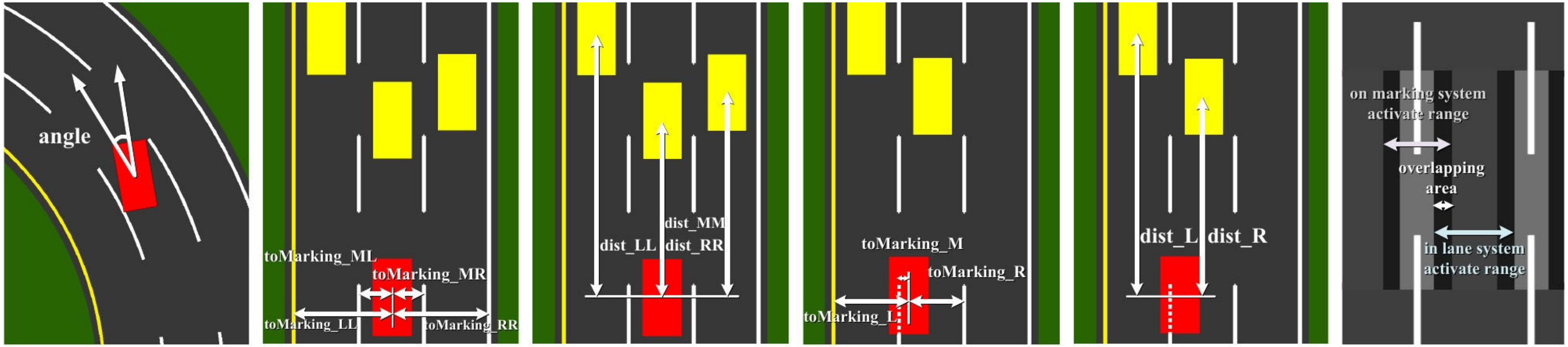
Direct Perception



Idea:

- **Hybrid model** between imitation learning and modular pipelines
- Learn to predict interpretable **low-dimensional** intermediate representation
- **Decouple** perception from planning and control
- Allows to exploit **classical** controllers or **learned** controllers (or hybrids)

Direct Perception for Autonomous Driving

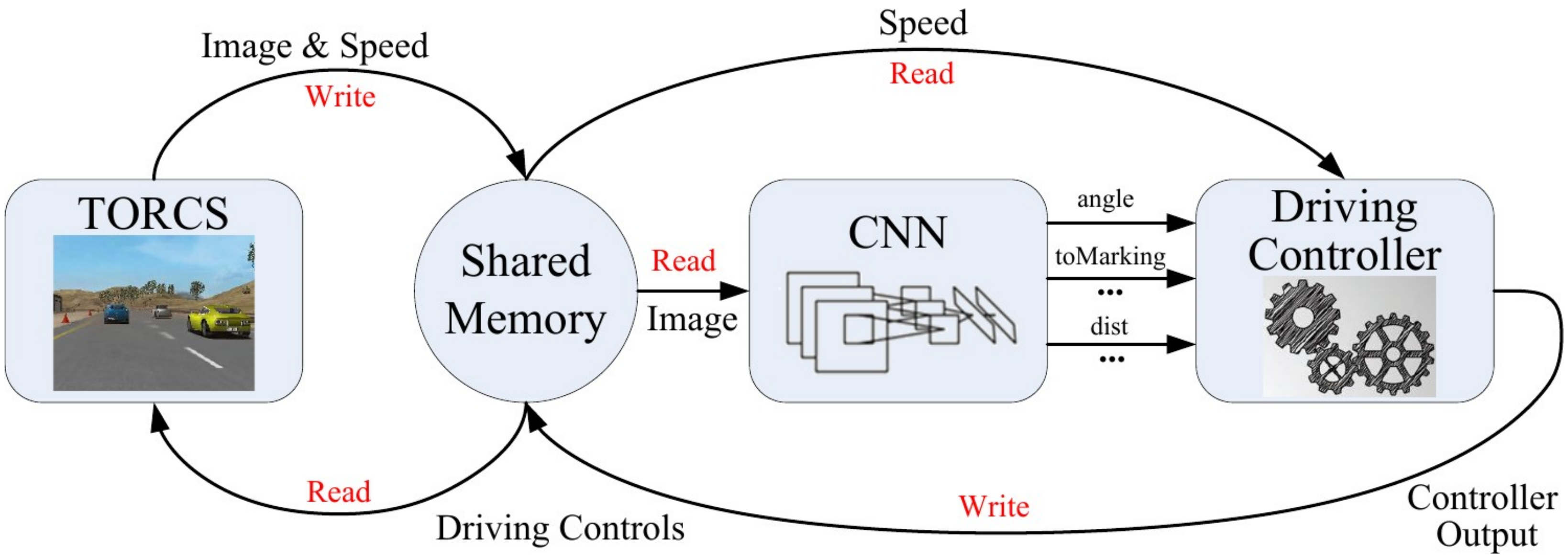


(a) angle (b) in lane: toMarking (c) in lane: dist (d) on mark.: toMarking (e) on marking: dist (f) overlapping area

Affordances:

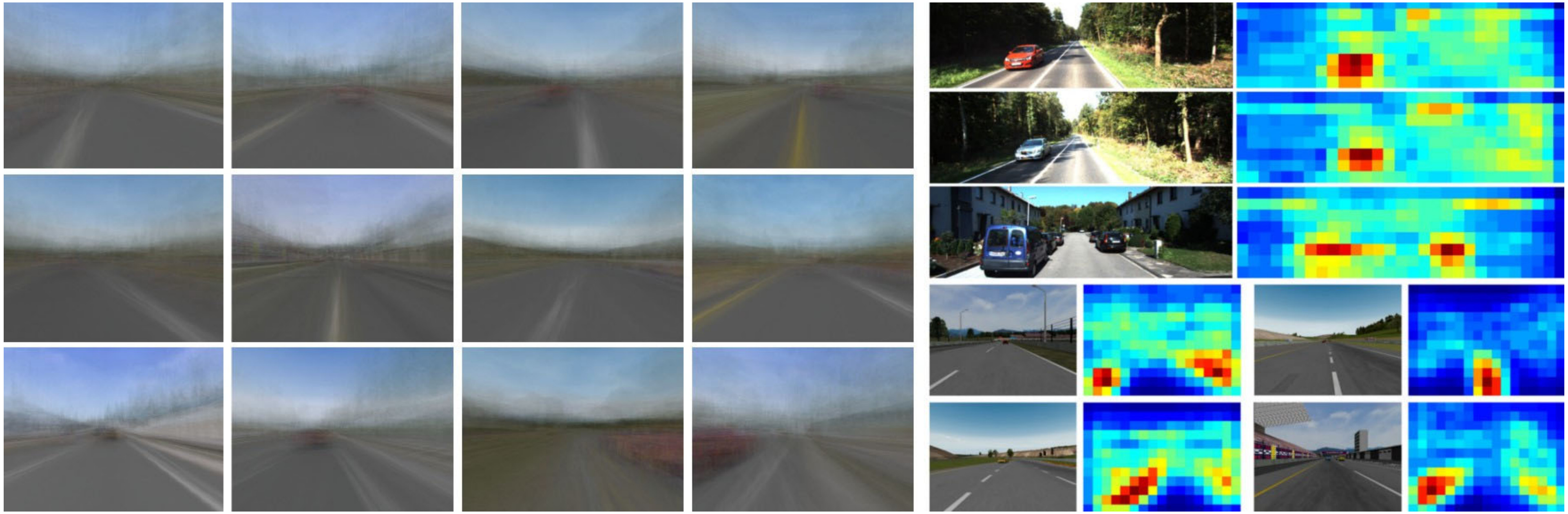
- **Attributes** of the environment which **limit space of actions** [Gibson, 1966]
- In this case: 13 affordances

Direct Perception for Autonomous Driving



- TORCS Simulator: Open source car racing game simulator
- Network: AlexNet (5 conv layers, 4 fully conn. layers), 73 output neurons
- Training: Affordance indicators trained with 2 loss

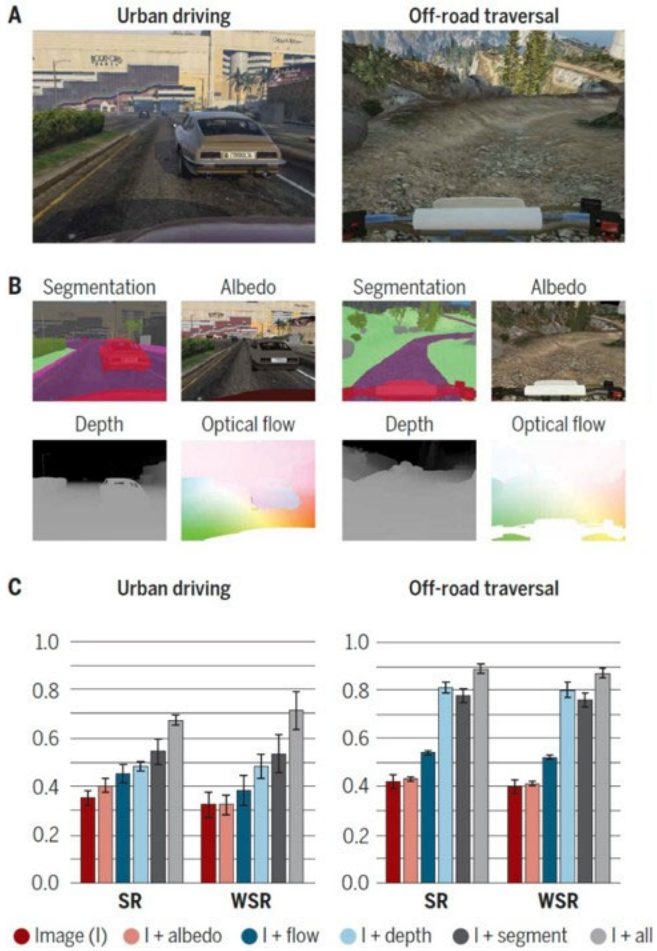
Network Visualization of the Perception Network



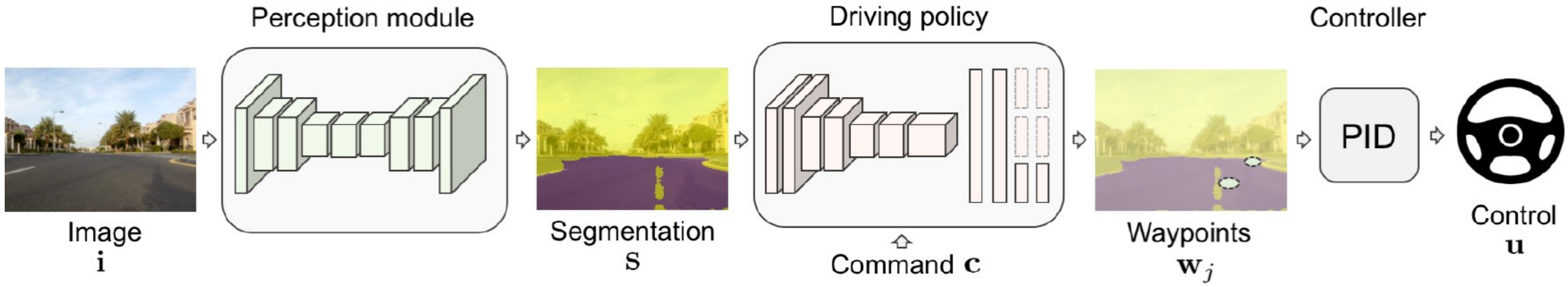
- Left: Averaged top 100 images activating a neuron in first fully connected layer
- Right: Maximal response of 4th conv. layer (note: focus on cars and markings)

Visual Abstractions as Generic Intermediate Representation

- Various visual abstraction as intermediate representation:
  - Segmentation, depth, normals, flow, albedo …
- Intermediate visual representations improve results
- Consistent gains across simulations / tasks
- Depth and semantic provide largest gains
- Better generalization performance
- Good visual abstraction: invariant, universal, data / label efficient
- Sematic segmentation:
  - Encodes task-relevant knowledge (e.g. road is drivable) and priors (e.g. grouping)
  - Can be processed with standard policy networks



Driving Policy Transfer



Problem:

- Driving policies learned in simulation often do not transfer well to the real world

Idea

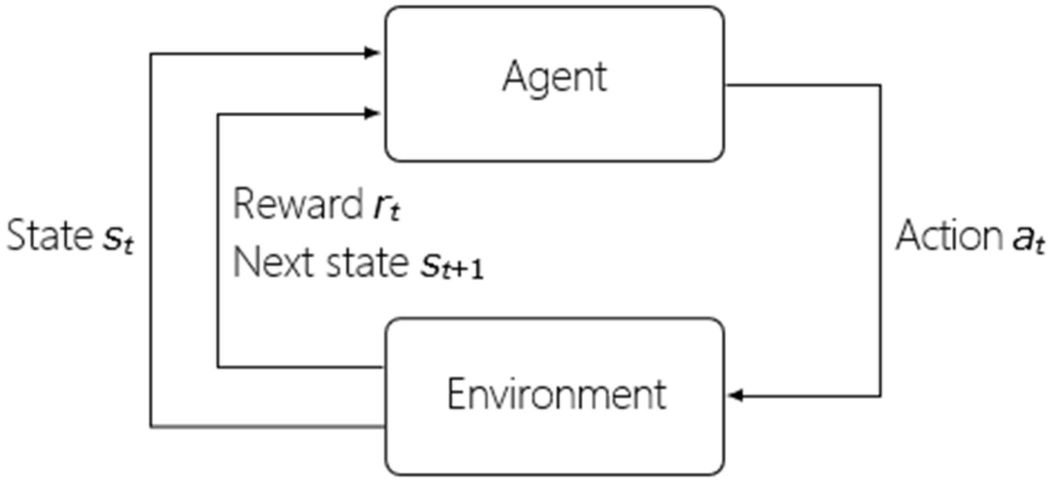
- Encapsulate driving policy such that it is not directly exposed to raw perceptual input or low-level control (input: semantic segmentation, output: waypoints)
- Allows for transferring driving policy without retraining or finetuning

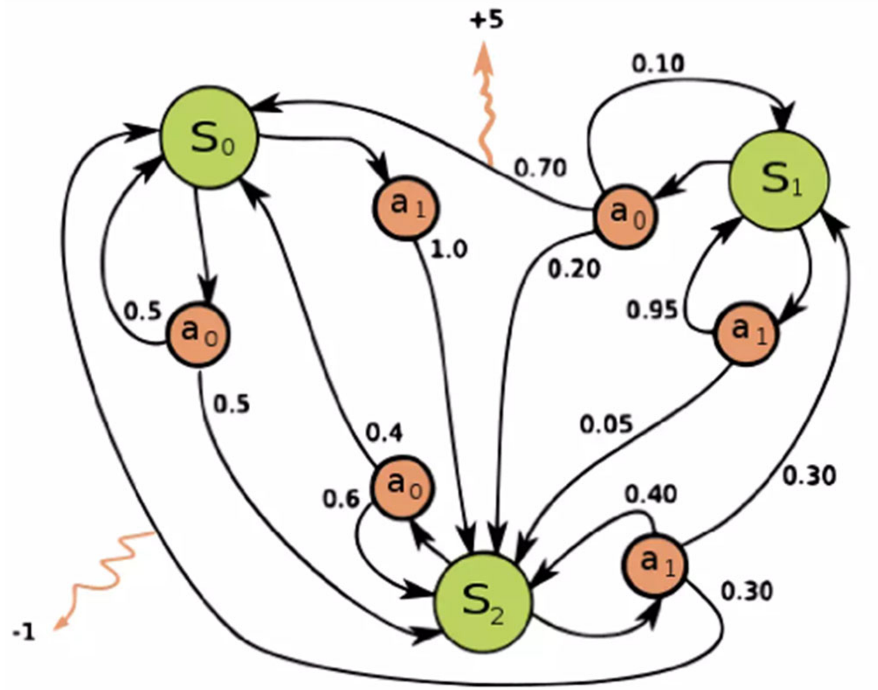
Autonomous Driving: Reinforcement Learning

Reinforcement Learning

- **So far:**
  - Supervised learning, lots of expert demonstrations required
  - Use of auxiliary, short-term loss functions
    - Imitation learning: per-frame loss on action
    - Direct perception: per-frame loss on affordance indicators
- **Can we:**
  - Learning of models based on the loss that we actually care about, e.g.:
    - Minimize time to target location
    - Minimize number of collisions
    - Minimize risk
    - Maximize comfort
    - etc.

Reinforcement Learning





- Agent observes environment state s_t at time t
- Agent sends action at a_t time t to the environment
- Environment returns the reward r_t and its new state s_{t+1} to the agent
- A RL problem can be formalized as a Markov Decision Process (MDP).

Learning to Drive in a Day

Real-world RL demo by Wayve:

- Deep Deterministic Policy Gradients with Prioritized Experience Replay
- Input: Single monocular image
- Action: Steering and speed
- Reward: Distance traveled without the safety driver taking control (requires no maps / localization)
- 4 Conv layers, 2 FC layers
- Only 35 training episodes

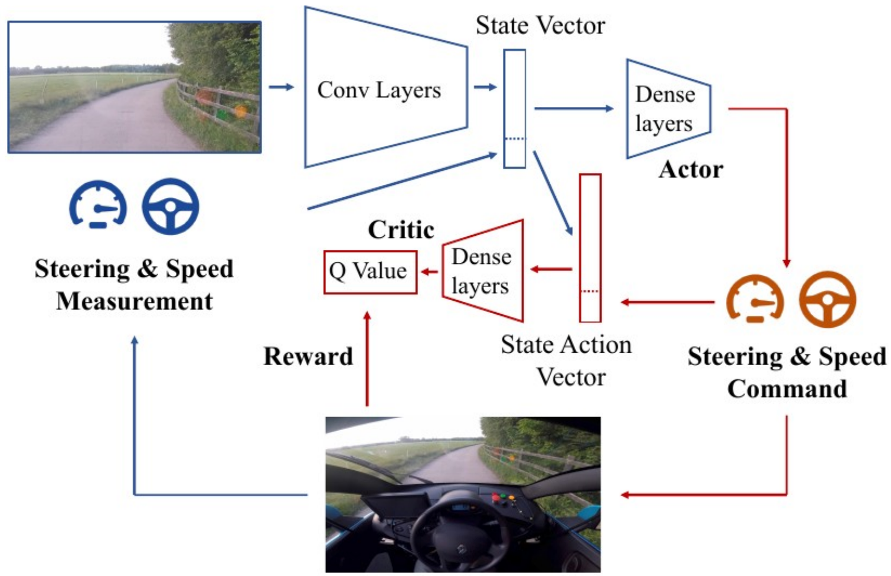


Fig. 1: We design a deep reinforcement learning algorithm for autonomous driving. This figure illustrates the actor-critic algorithm which we use to learn a policy and value function for driving. Our agent maximises the reward of distance travelled before intervention by a safety driver.

Multi-agent Problem

- Have to consider other participants
- Interaction:
  - Cooperative: all agents desire safety and efficiency
  - Competitive: each agent has its own goal
- Environment:
  - Non-deterministic
  - Partially observable
  - Dynamic

Conclusions

- Autonomous driving will reshape the future of transportation universally
- Technically we are not there yet, still many problems remains
- Artificial Intelligence is one of the main driving forces to make autonomous driving a reality
- There will be a long period of mixed traffic where autonomous vehicles should cope well with human-driven vehicles and other human traffic participants
- There are also legal, ethical, social, security issues to be handled

Thanks