certified control: a new safety architecture

Principal Investigators:
Daniel Jackson & Armando Solar-Lezama

Team Members:
Nikos Arechiga, Jeff Chow, **Jonathan DeCastro**, Uriel Guajardo, **Soonho Kong**,
Dimitris Koutentakis, Angela Leong, Geoffrey Litt, Valerie Richmond, Mike Wang & Xin Zhang

TRI Liaison:
Soonho Kong

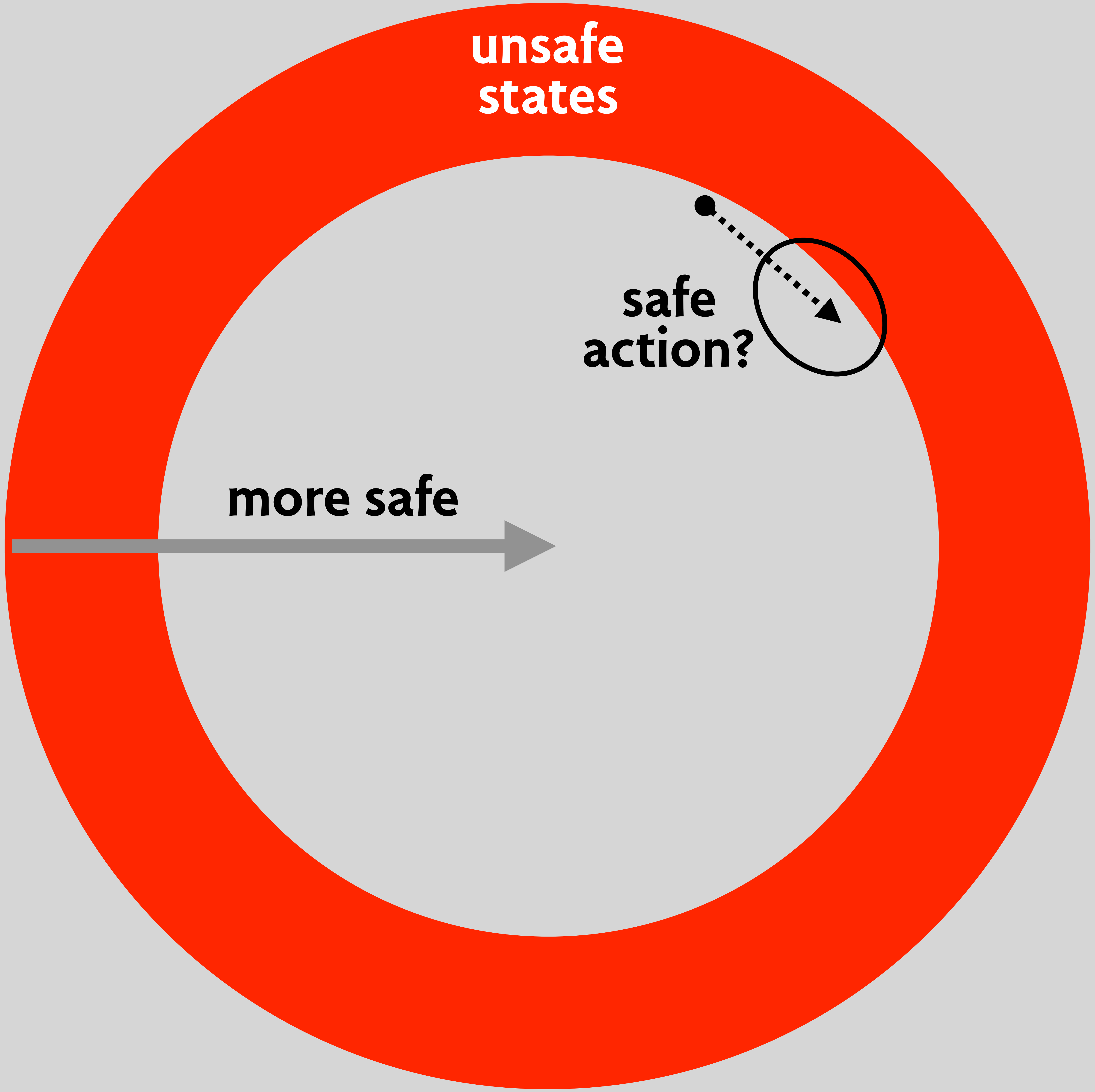
Quarterly Review Meeting
Toyota-CSAIL Joint Research Center
December 5, 2019

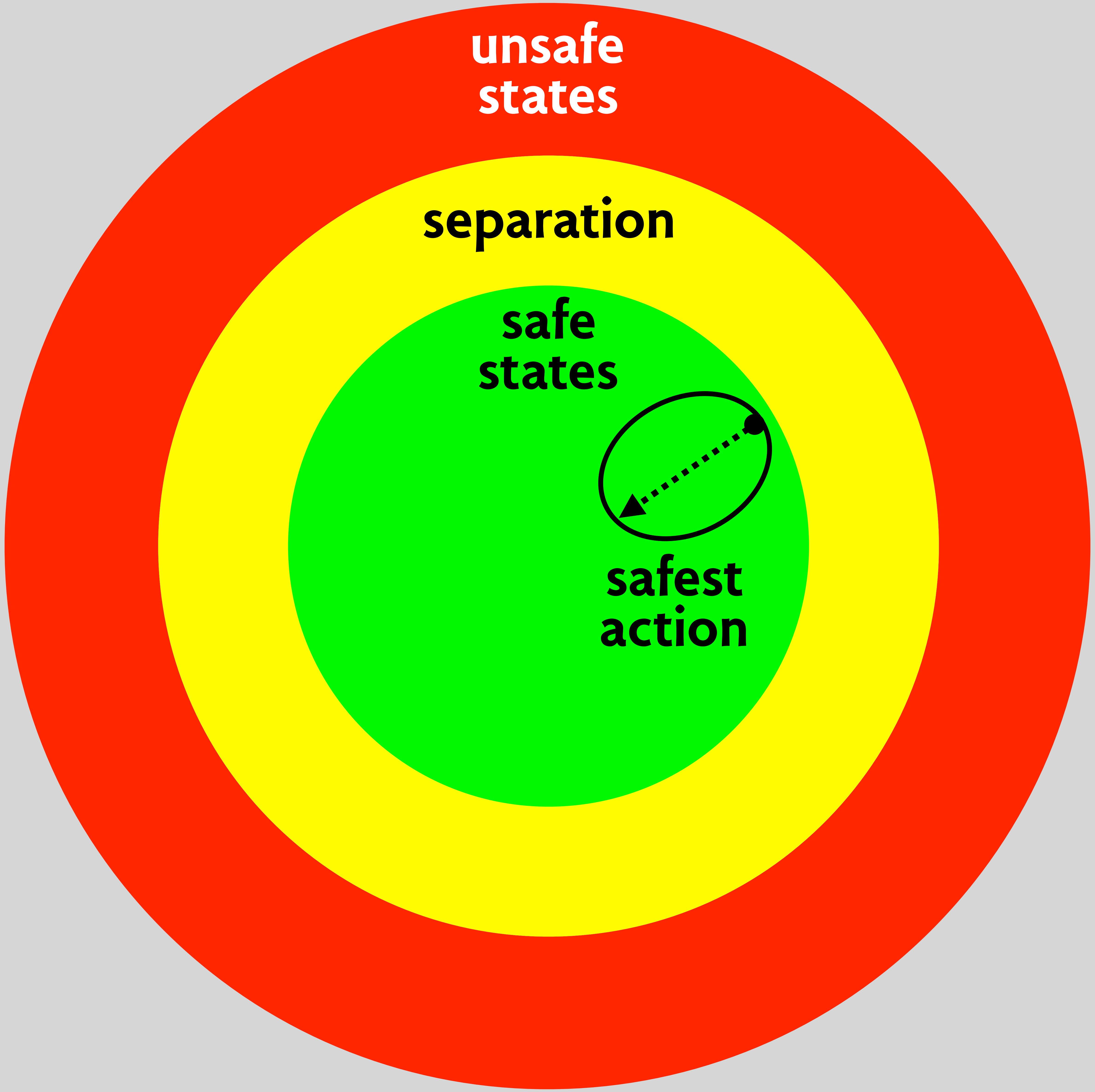
TOYOTA
RESEARCH INSTITUTE

Massachusetts
Institute of
Technology

CSAIL
MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

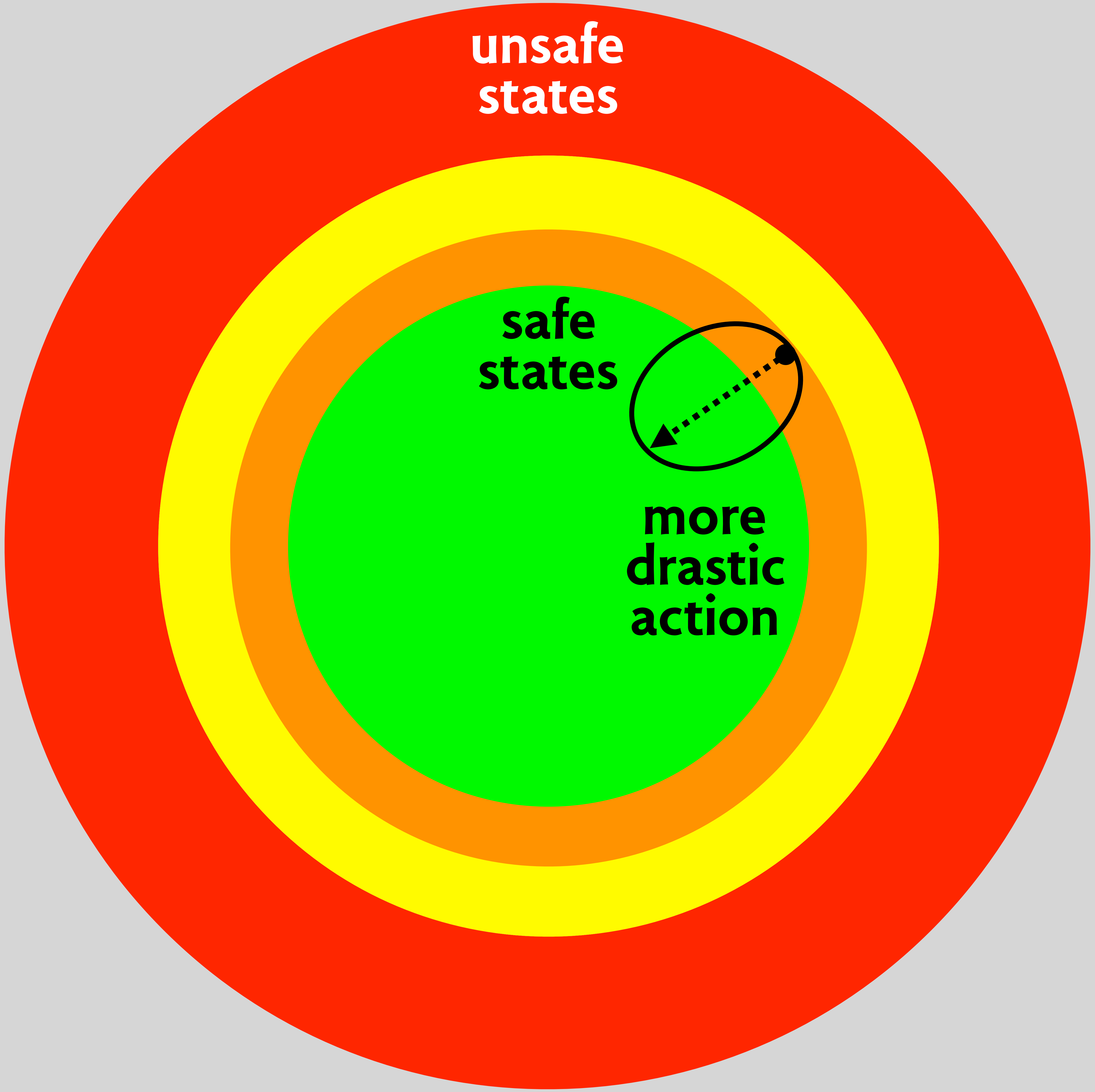
part 1: how to design a safe controller

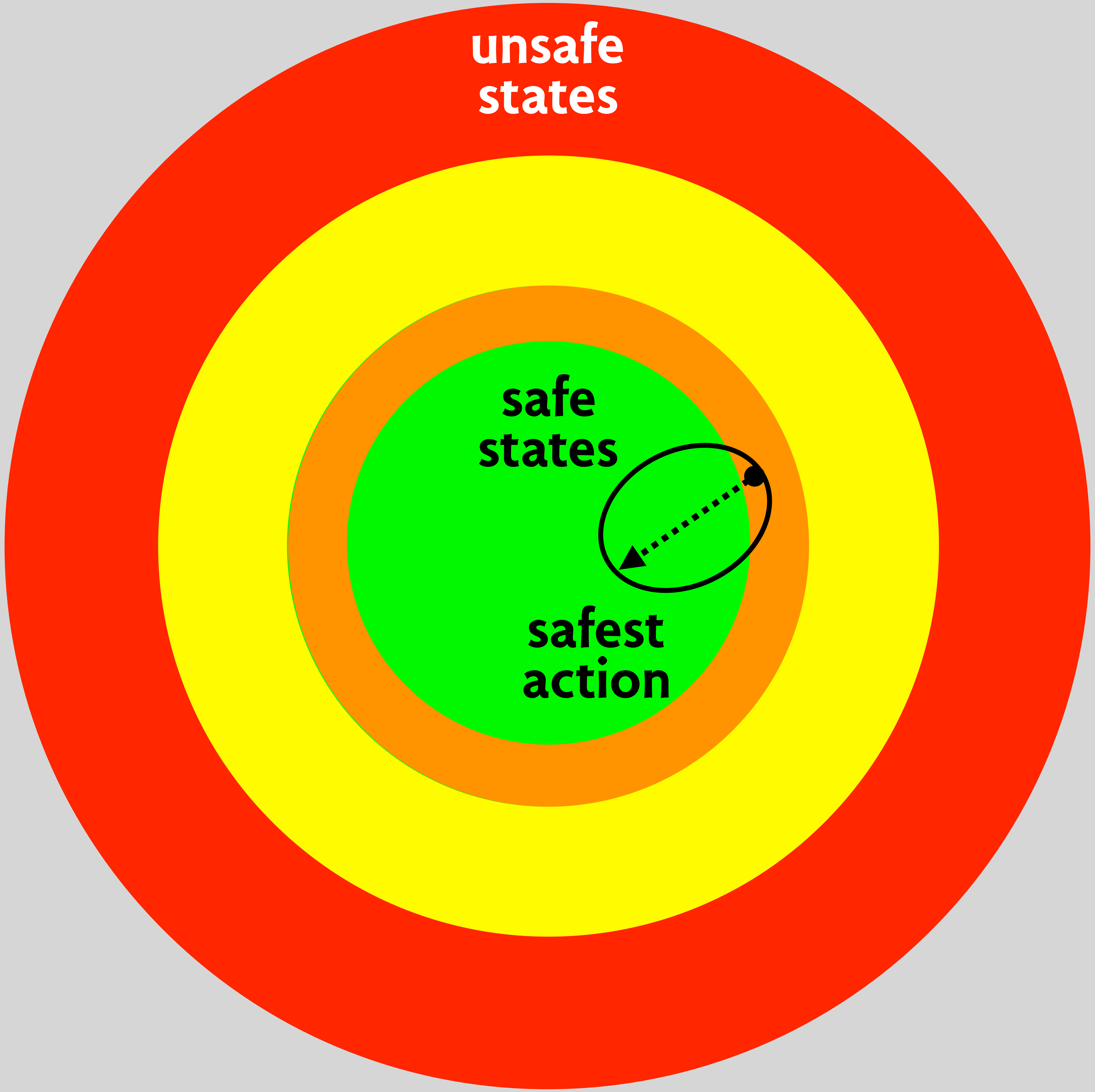

unsafe
states
safe
action?
more safe



but what if mechanism is too complex to verify?

part 2: classic interlocks


unsafe states
safe states
more drastic action



interlock only needs to intervene, so it can be verifiable

suppose state perception depends on LiDAR

in snow, controller does complex (ie, error-prone) filtering of snow

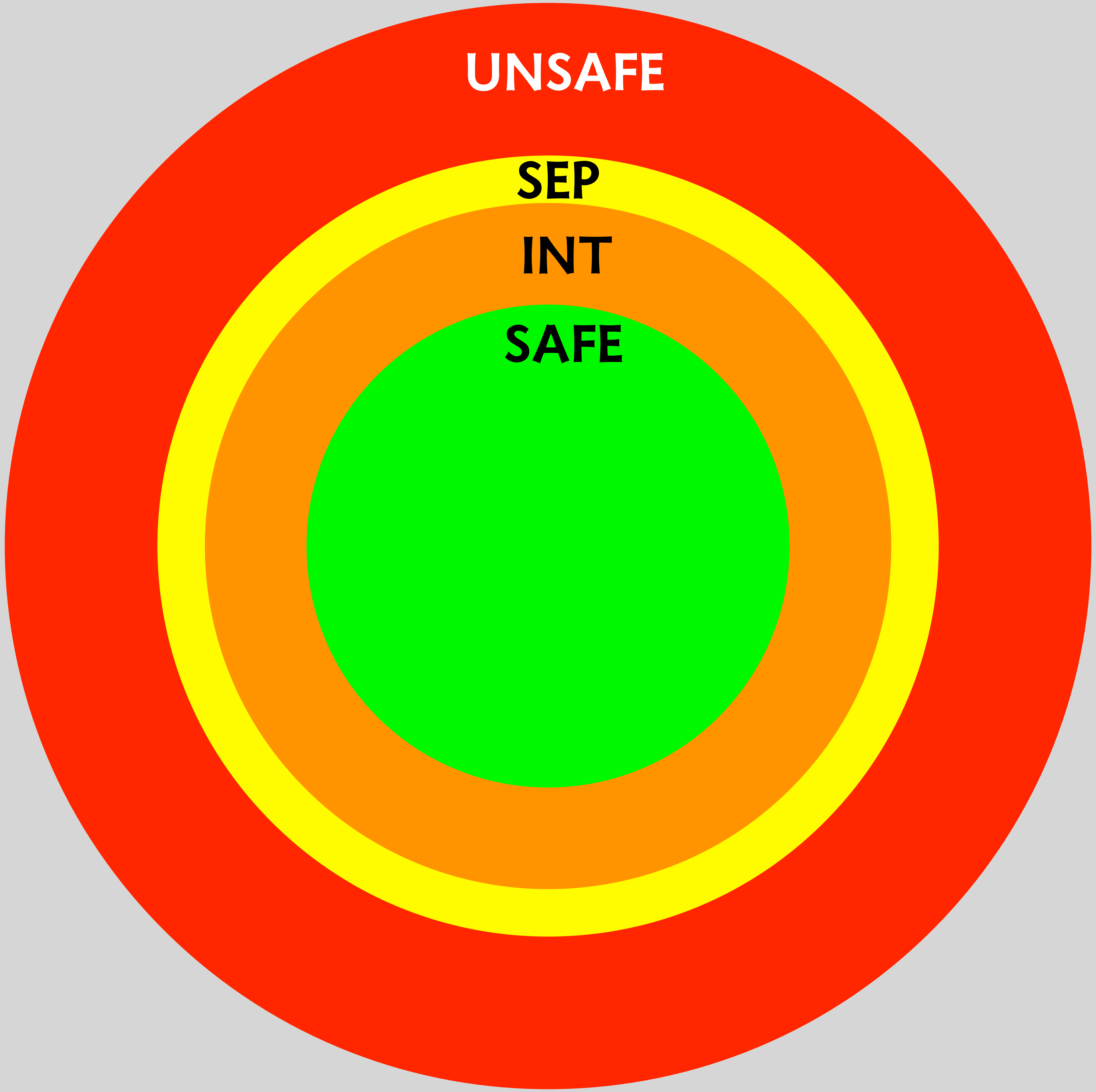
option 1: interlock does filtering too

then can’t verify interlock either

option 2: interlock does something simpler

then interlock will intervene by design

3 interlock properties: pick 2



sound
intervene only on failure
SAFE ∩ INT = ∅

complete
interlock prevents accident
UNSAFE ∩ INT = ∅
and interlock can maintain INT

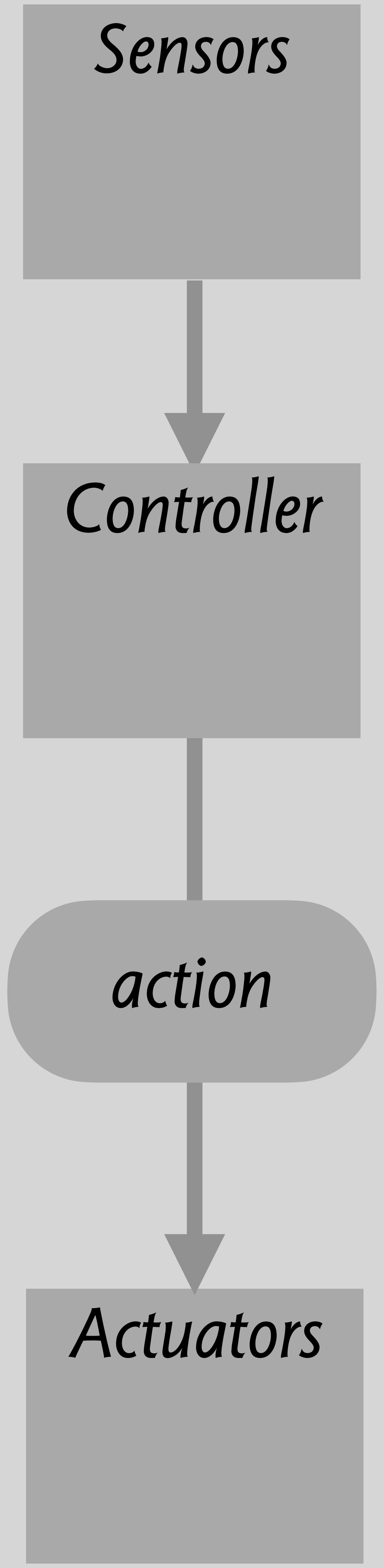
AEB drops

robust
interlock check is simple
s ∈ INT is verifiable

RSS drops

part 3: certified control

no interlock



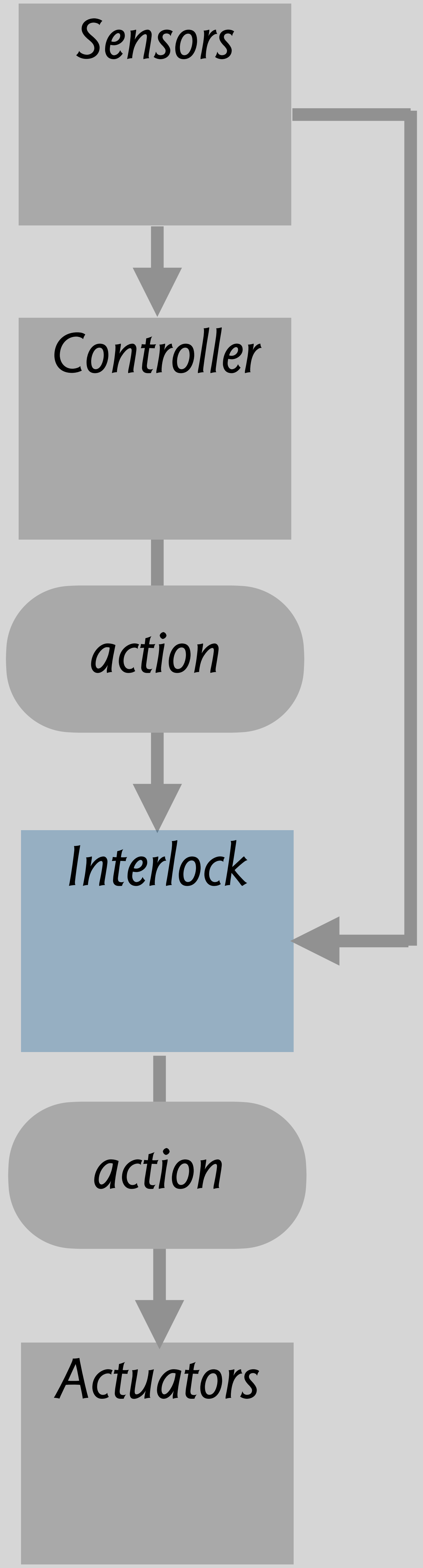
Sensors

Controller

action

Actuators

classic interlock



Sensors

Controller

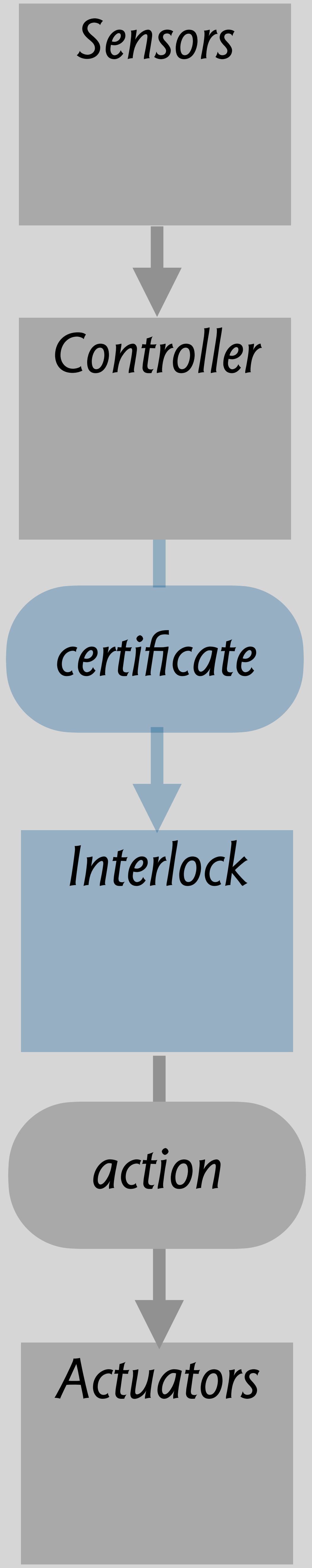
action

Interlock

action

Actuators

certified control



Sensors

Controller

certificate

Interlock

action

Actuators

the essence of certified control

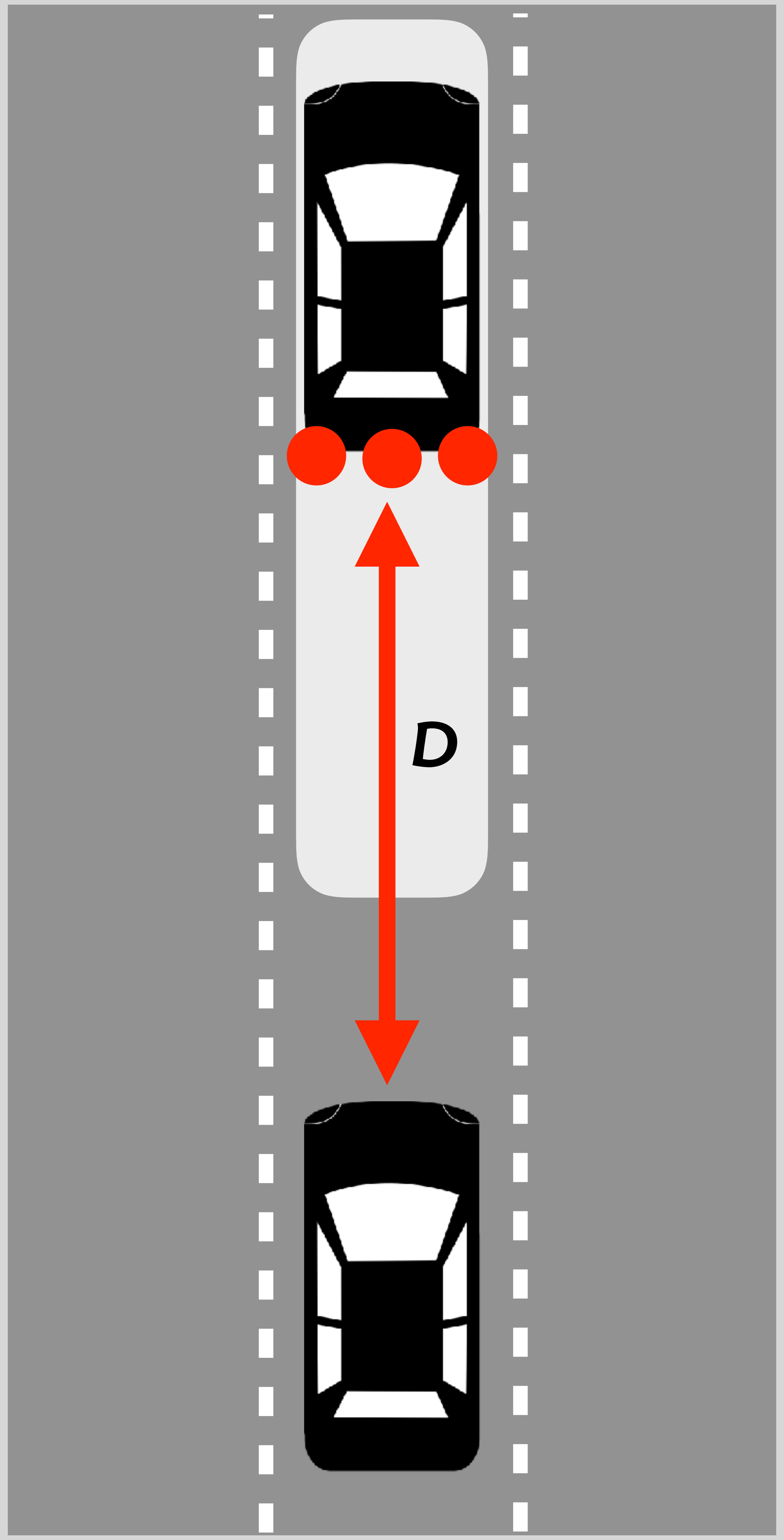
controller can always generate certificate
$\forall s \cdot\ s \in \mathrm{SAFE} \Rightarrow \exists A, i \cdot \mathrm{CERT}(s, i, A)$

runtime dependability case

if certificate holds, then guarantees no crashes
$\forall s{:}\ \mathrm{SAFE}, i, A \cdot\ \mathrm{CERT}(s, i, A) \Rightarrow \forall s' \cdot\ A(s,s') \Rightarrow s' \in \mathrm{SAFE}$

agreed upon at design time

example: certificate for continuing ahead



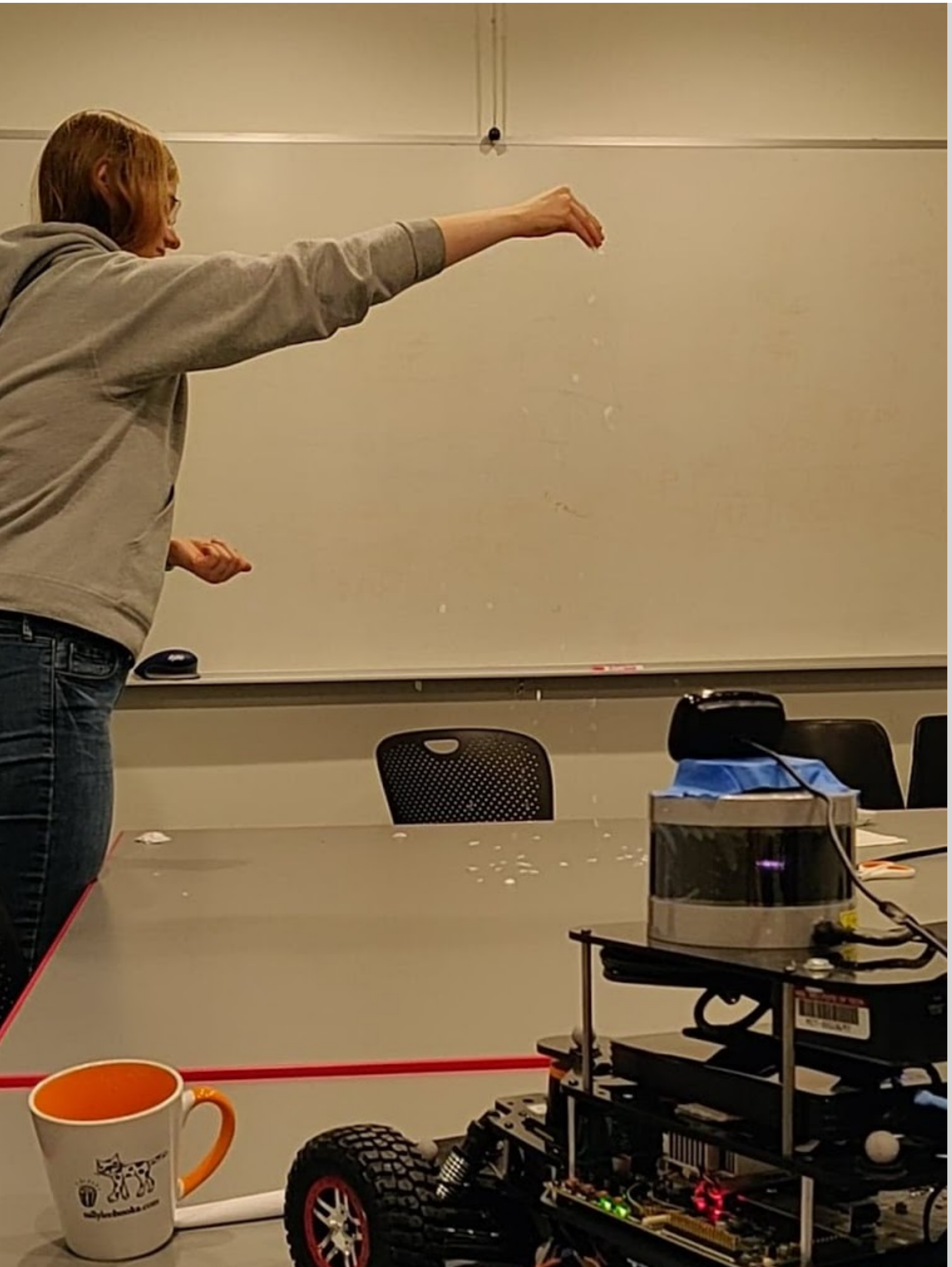
elements of the certificate

action **A**: continue ahead without decreasing speed
state **s**: 3 LIDAR readings L[0..2] (signed by LIDAR unit)
ego car velocity V (signed by velocity unit)
interpretation **i**: a distance D

checking CERT(s,i,A)

authenticates sensor data using public keys of sensors
checks L[0..2] lie on a straight line a distance D ahead
checks D > minimum separation at velocity V

a snow experiment



Controller

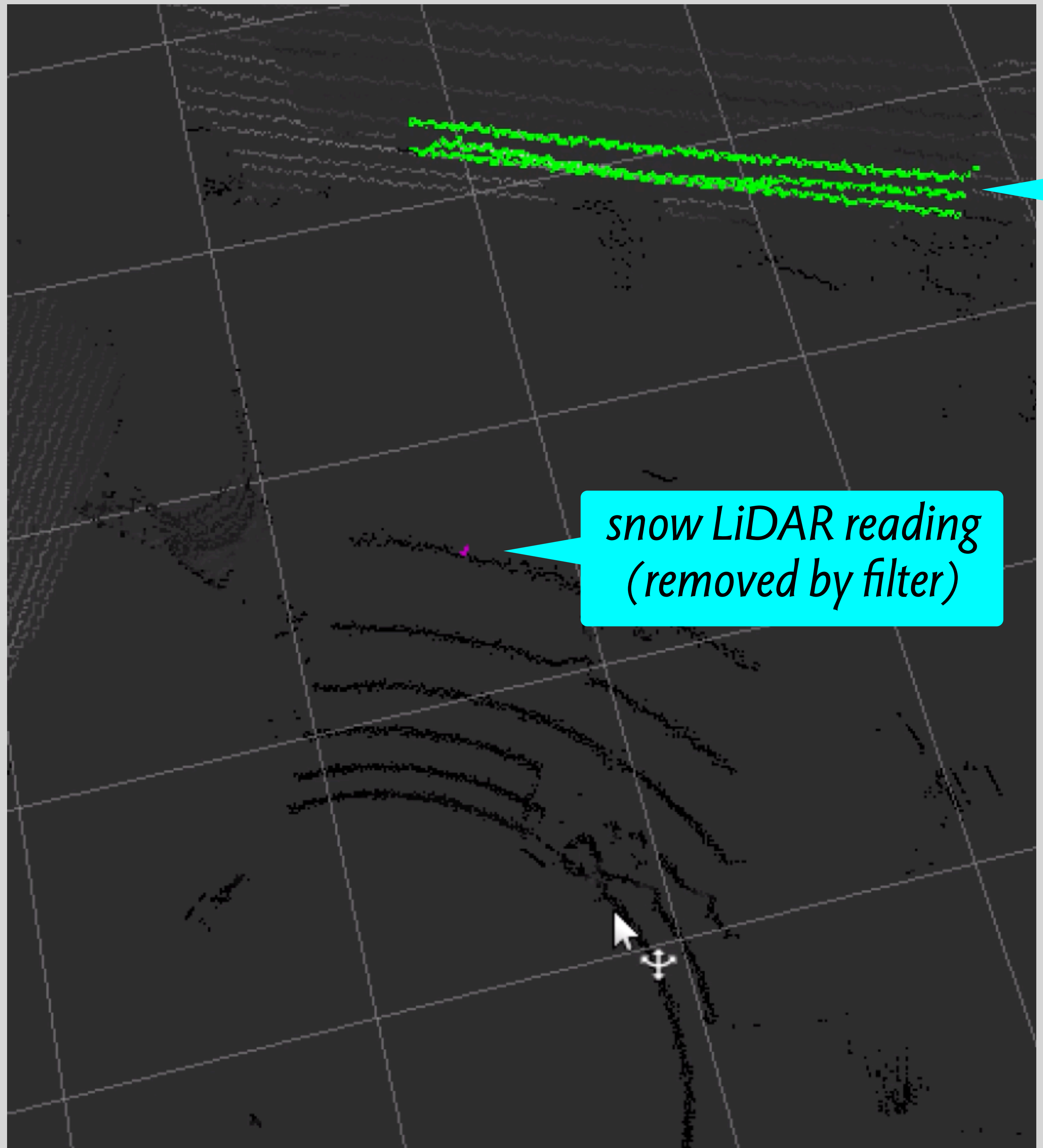
—**Filter LiDAR points** using 3D outlier detection* with K-d tree to remove snow

—**Generate certificate** of array of remaining LiDAR points at distance

Interlock

Check points in certificate are sufficiently close together and cover lane

**De-noising of Lidar Point Clouds Corrupted by Snowfall. Nicholas Charron, Stephen Phillips and Steven L. Waslander. Fifteenth Conference on Computer and Robot Vision (CRV 2018)*

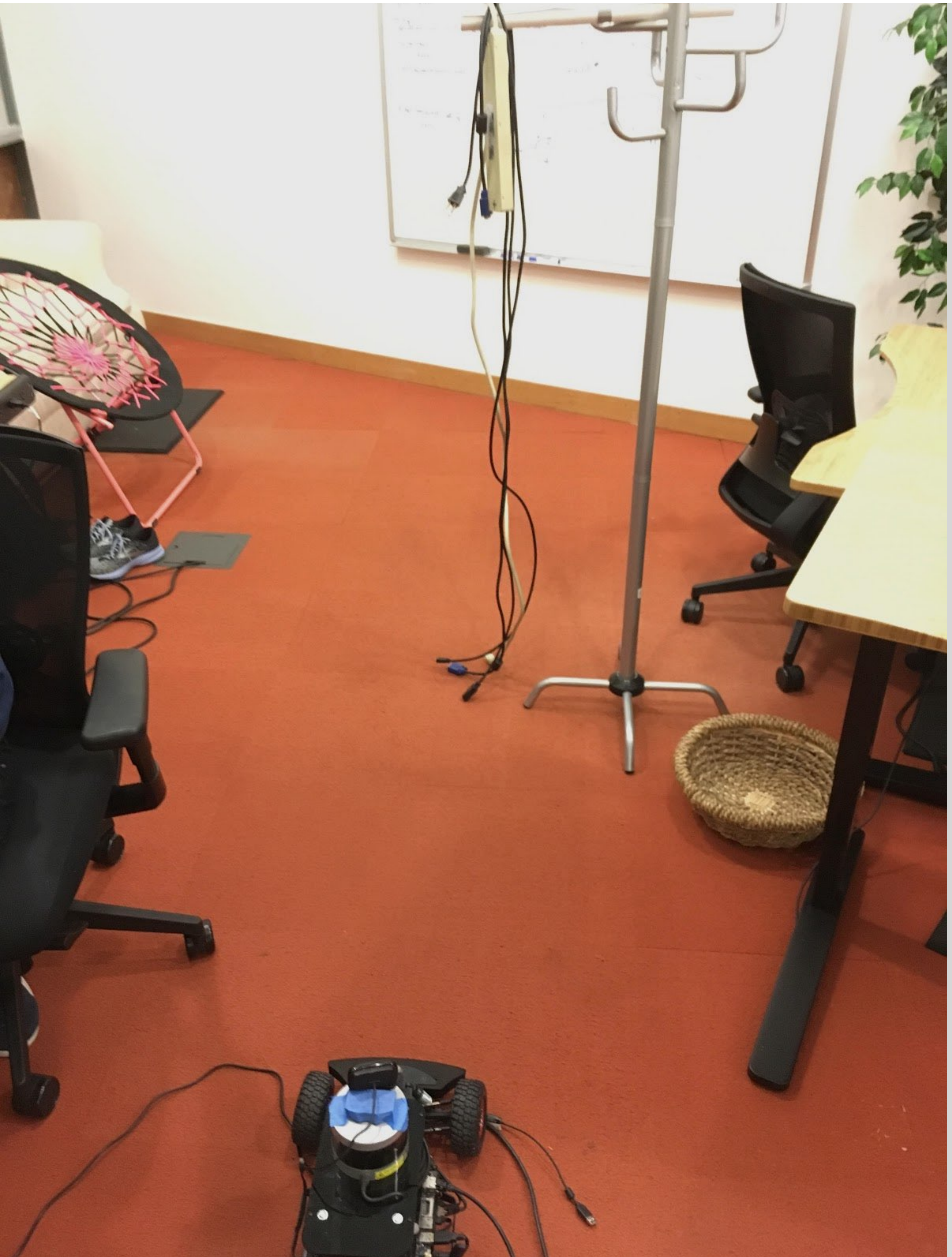


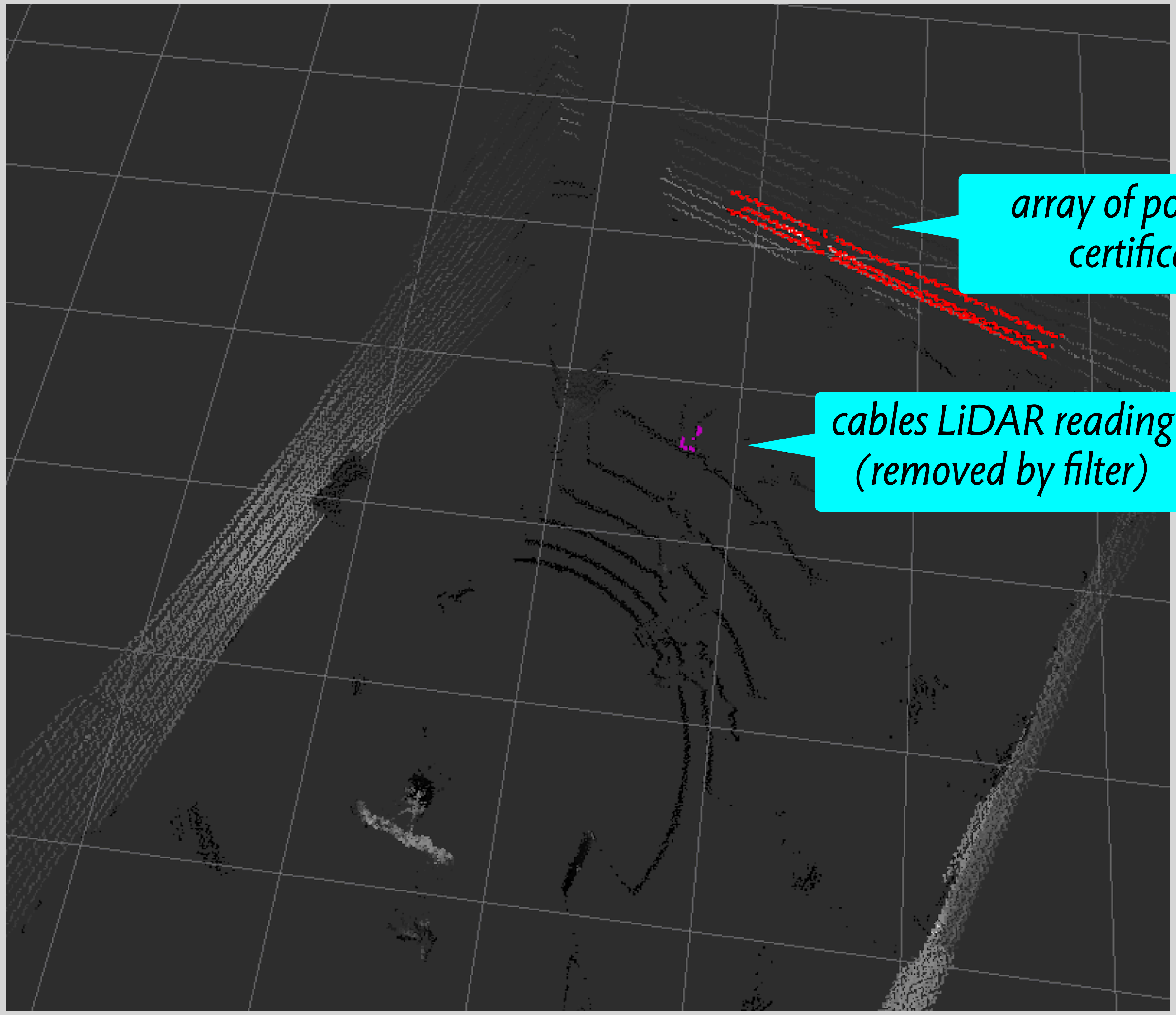
array of points in certificate

snow LiDAR reading (removed by filter)

✔ passes check

what about other small objects?





array of points in certificate

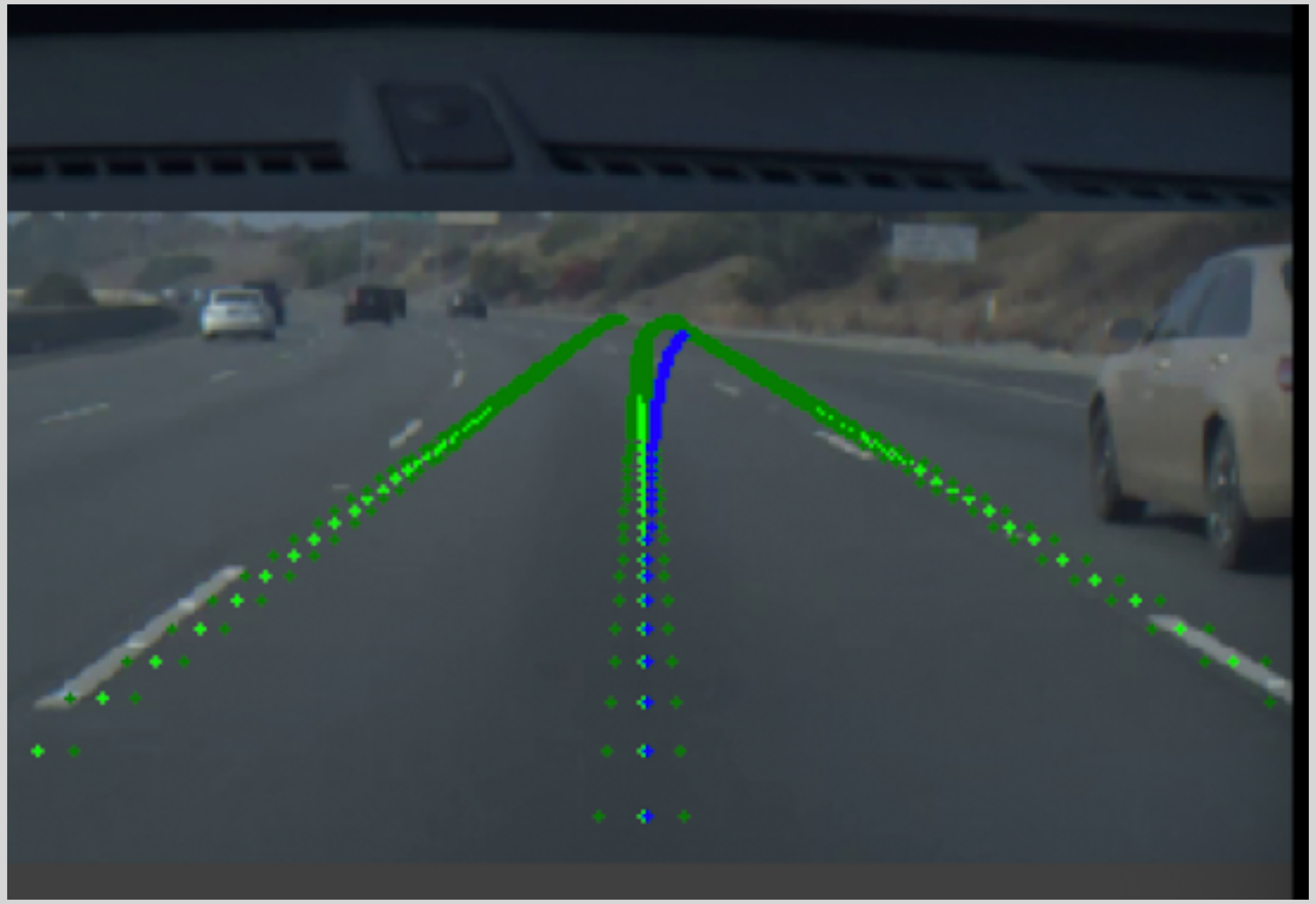
cables LiDAR reading (removed by filter)

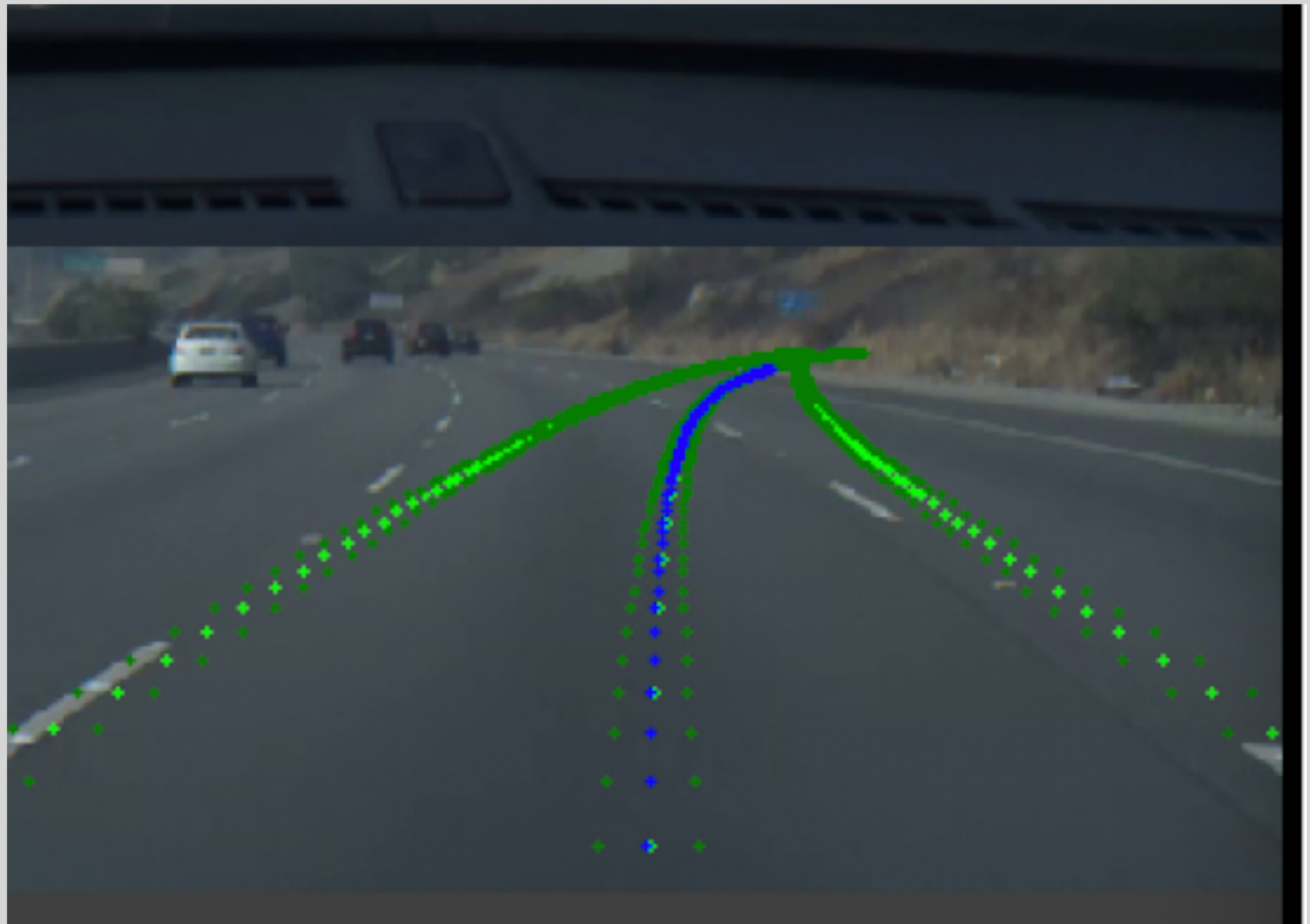
✗ fails check

certified control handles snow filtering for obstacle detection
but what about lane following? no pixels to pick like the LiDAR points

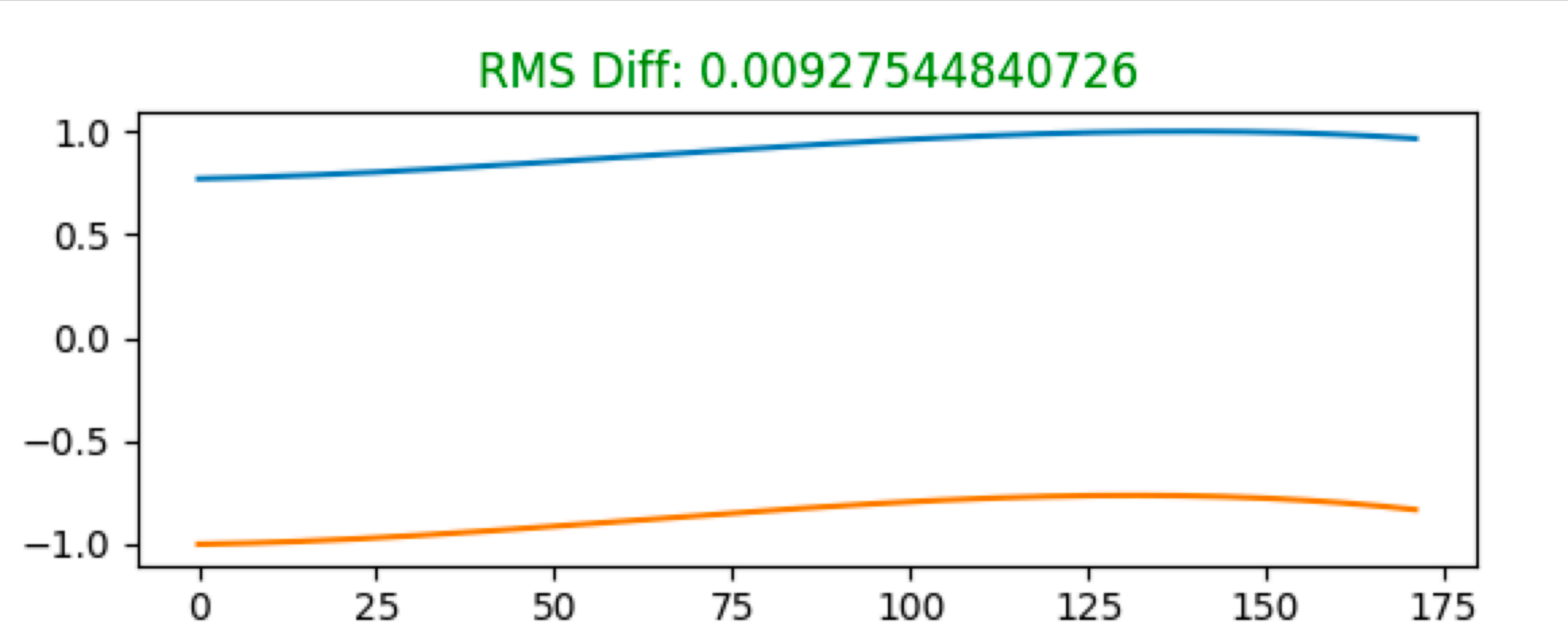
part 4: checking lane lines

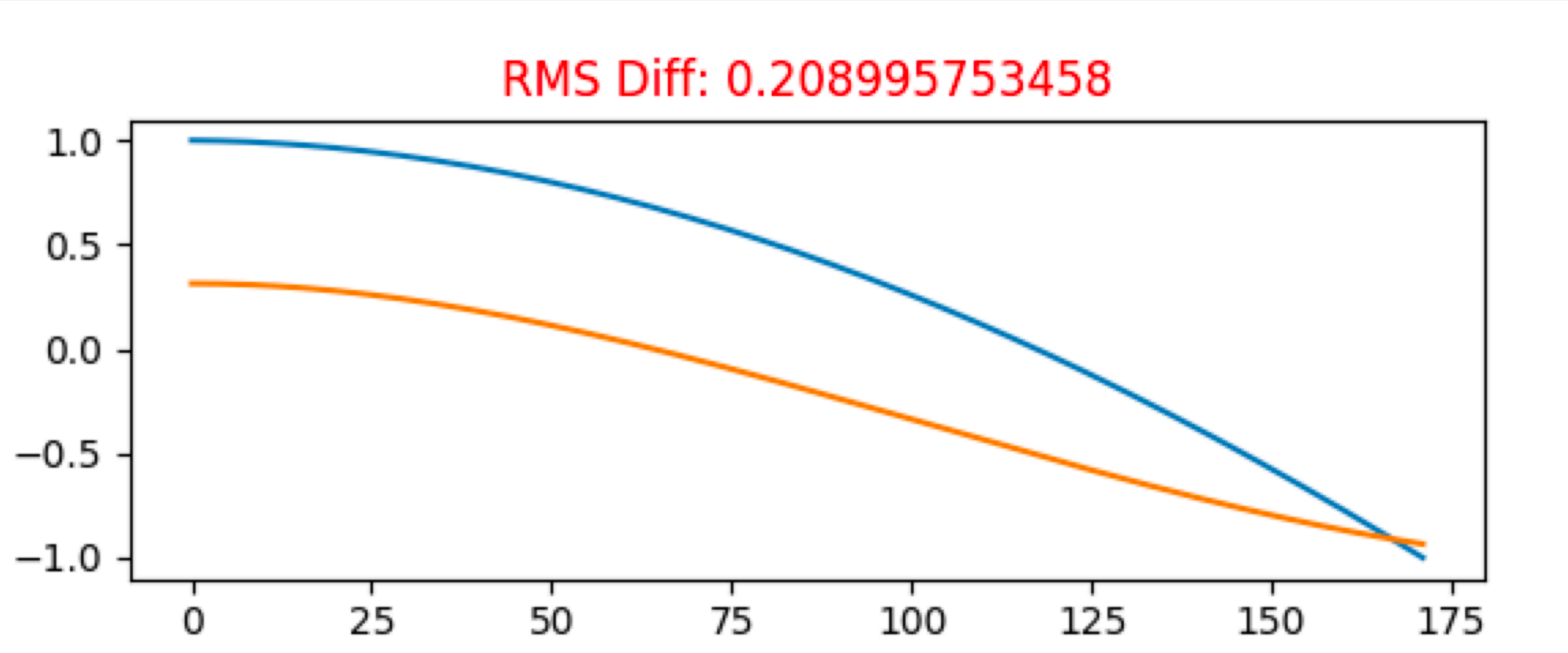
check #1: lane has the right geometry



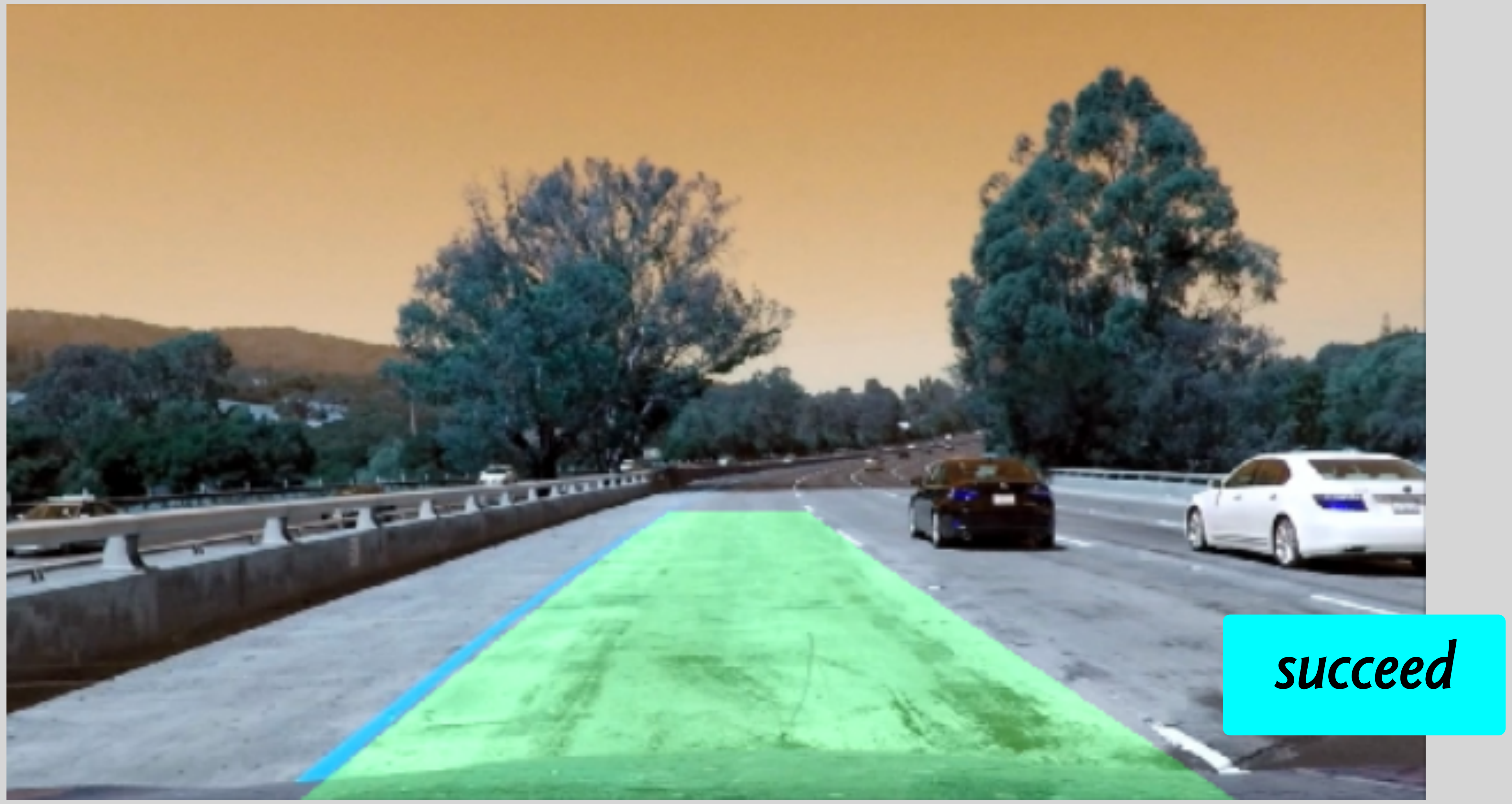


steps:
scale line segments to [-1, +1]
shift lines together by average distance
calculate root mean square difference

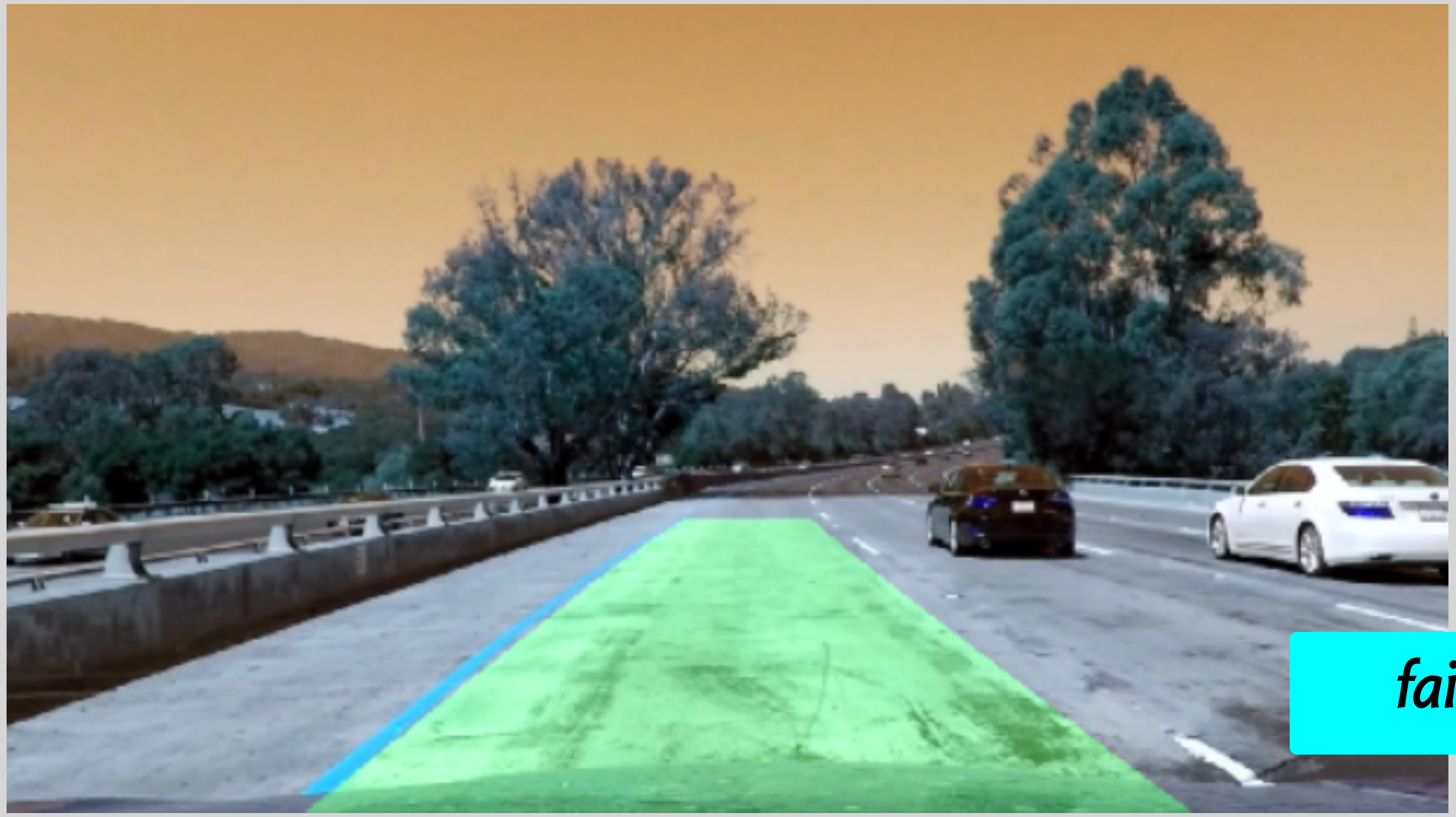




check #2: conformance to image



succeed



fail

steps:

detect markers with filters/edge detection

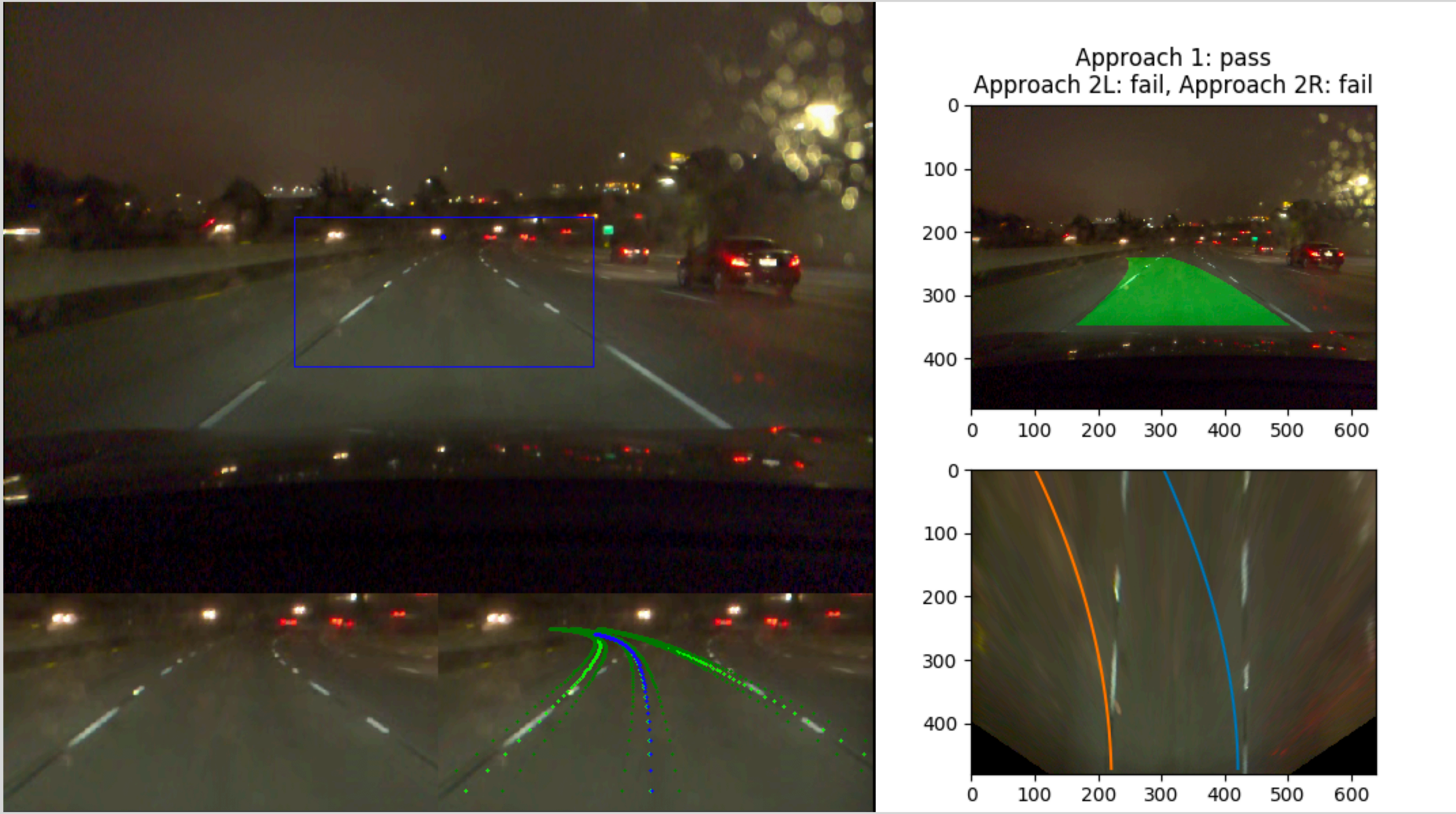
transform to bird's eye view

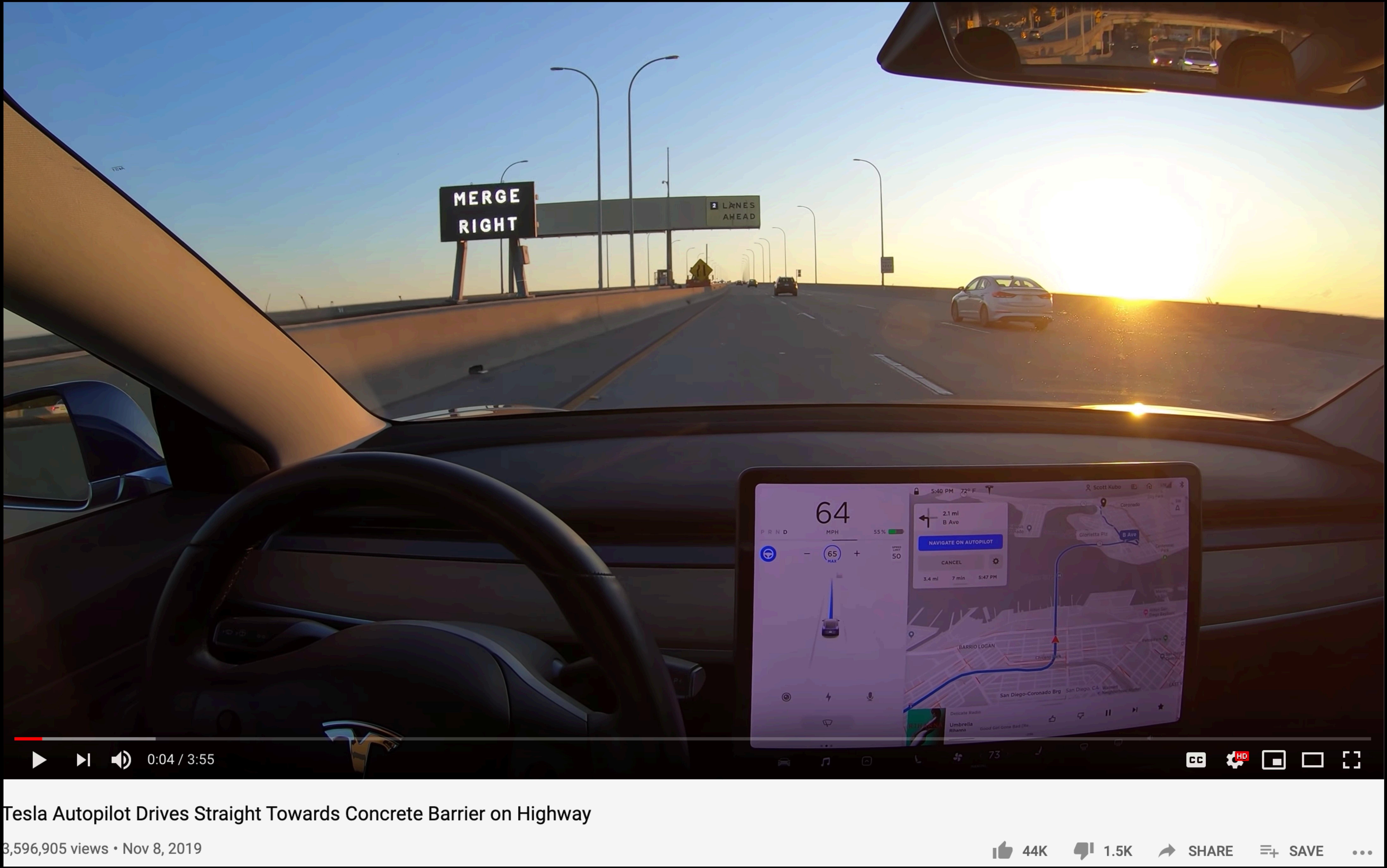
convolve with purported lanes, left & right

try solid line first, then dashed line

apply thresholds to decide if match

experiment: open pilot sample video





Tesla Autopilot Drives Straight Towards Concrete Barrier on Highway

3,596,905 views • Nov 8, 2019

tesla accident experiment

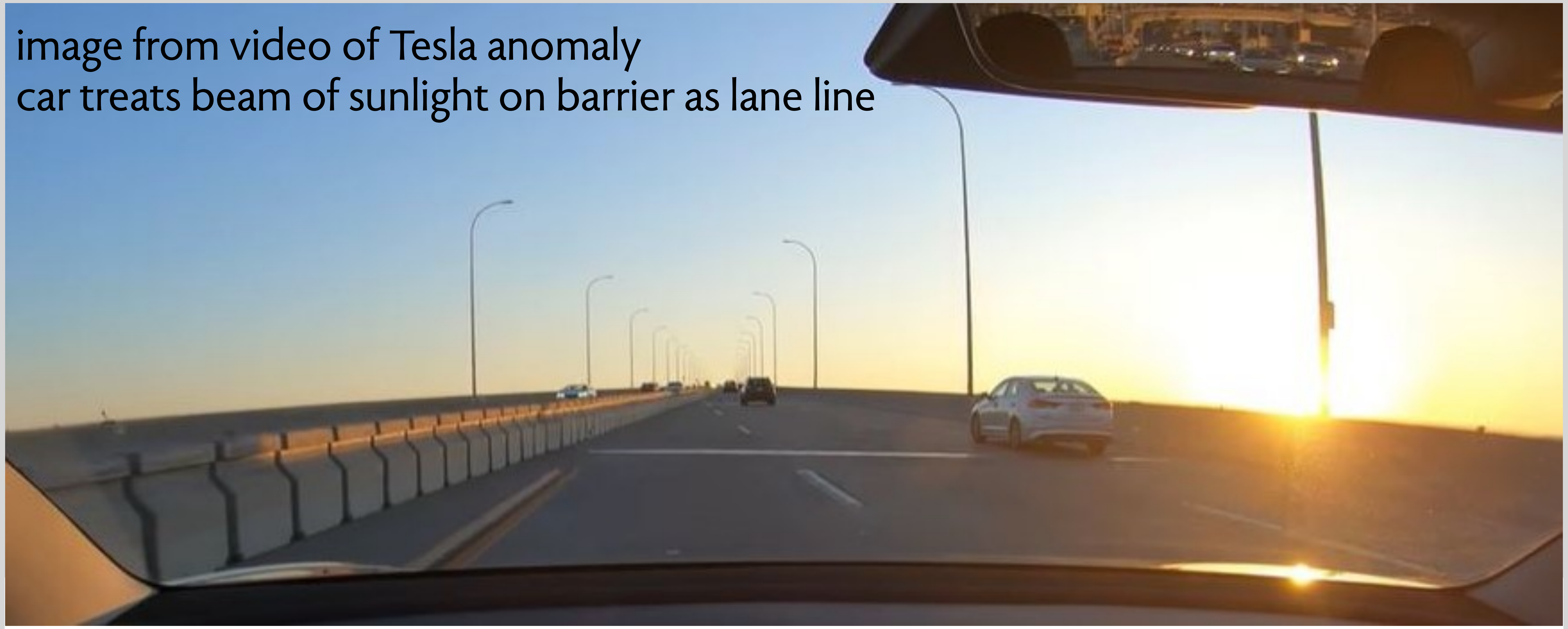
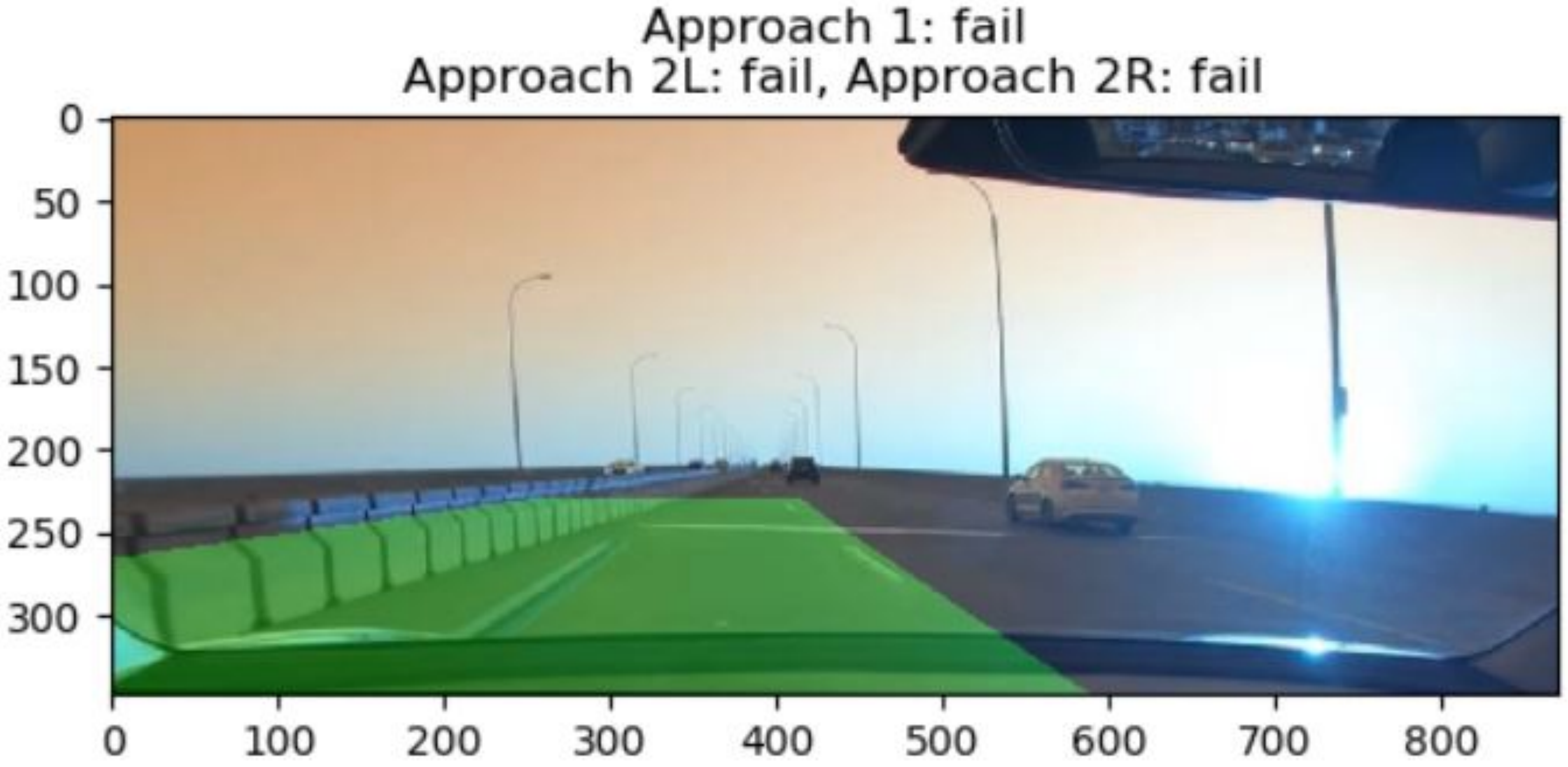


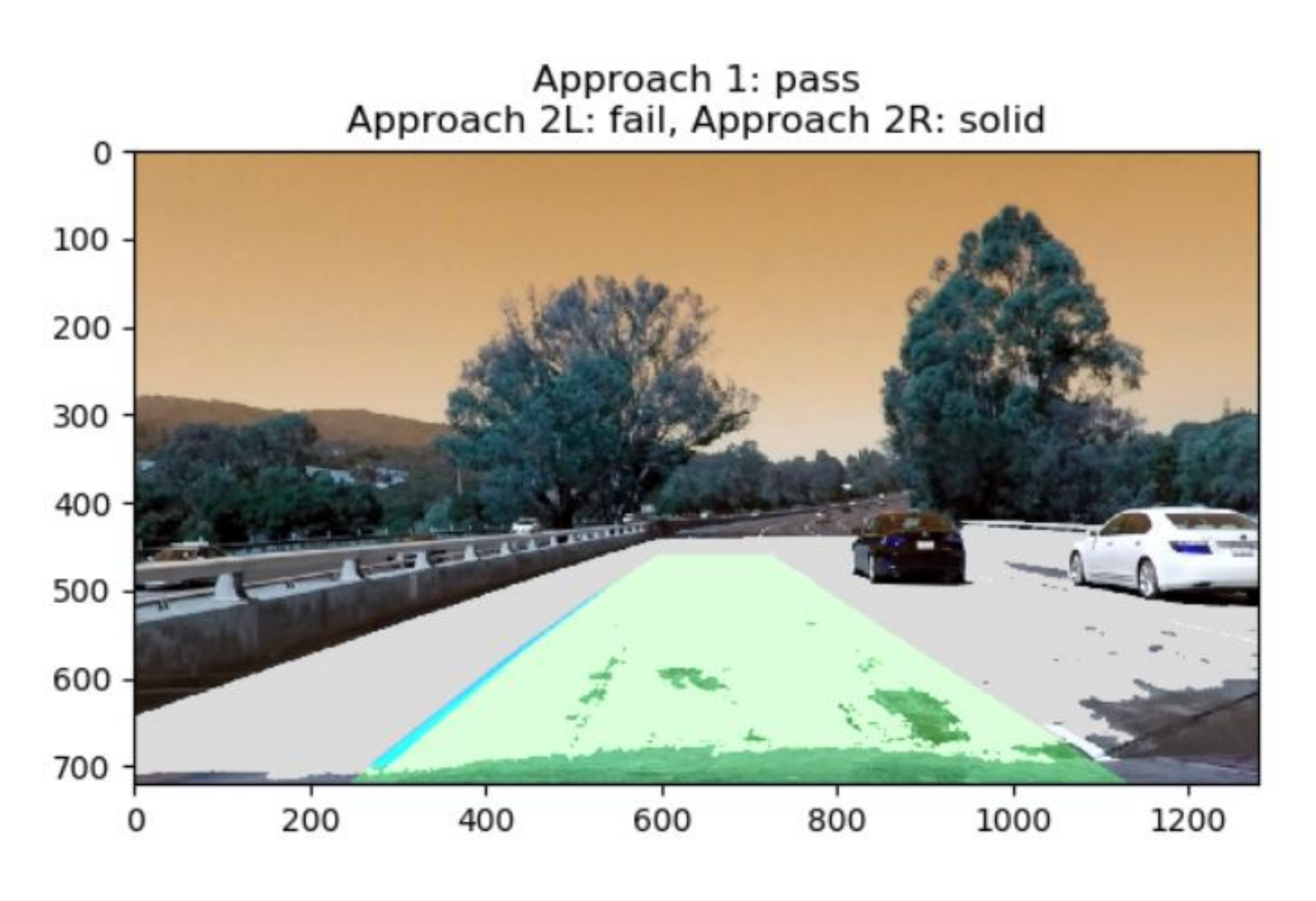
image from video of Tesla anomaly
car treats beam of sunlight on barrier as lane line



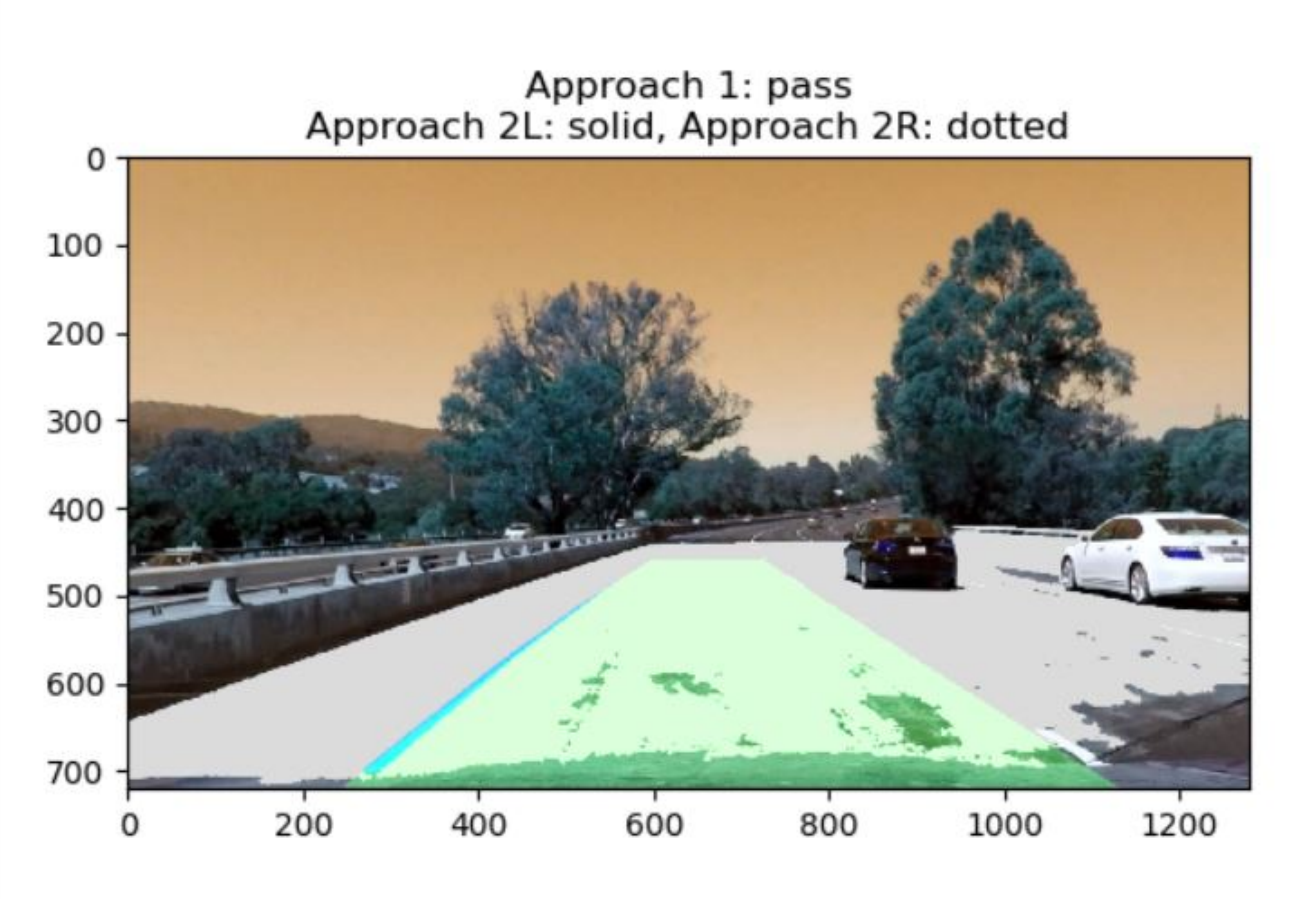
Approach 1: fail
Approach 2L: fail, Approach 2R: fail

simulating adverse lighting

modified road image to make road surface light grey
now controller needs to apply color filter to find lane line
can pass color filter parameters to interlock

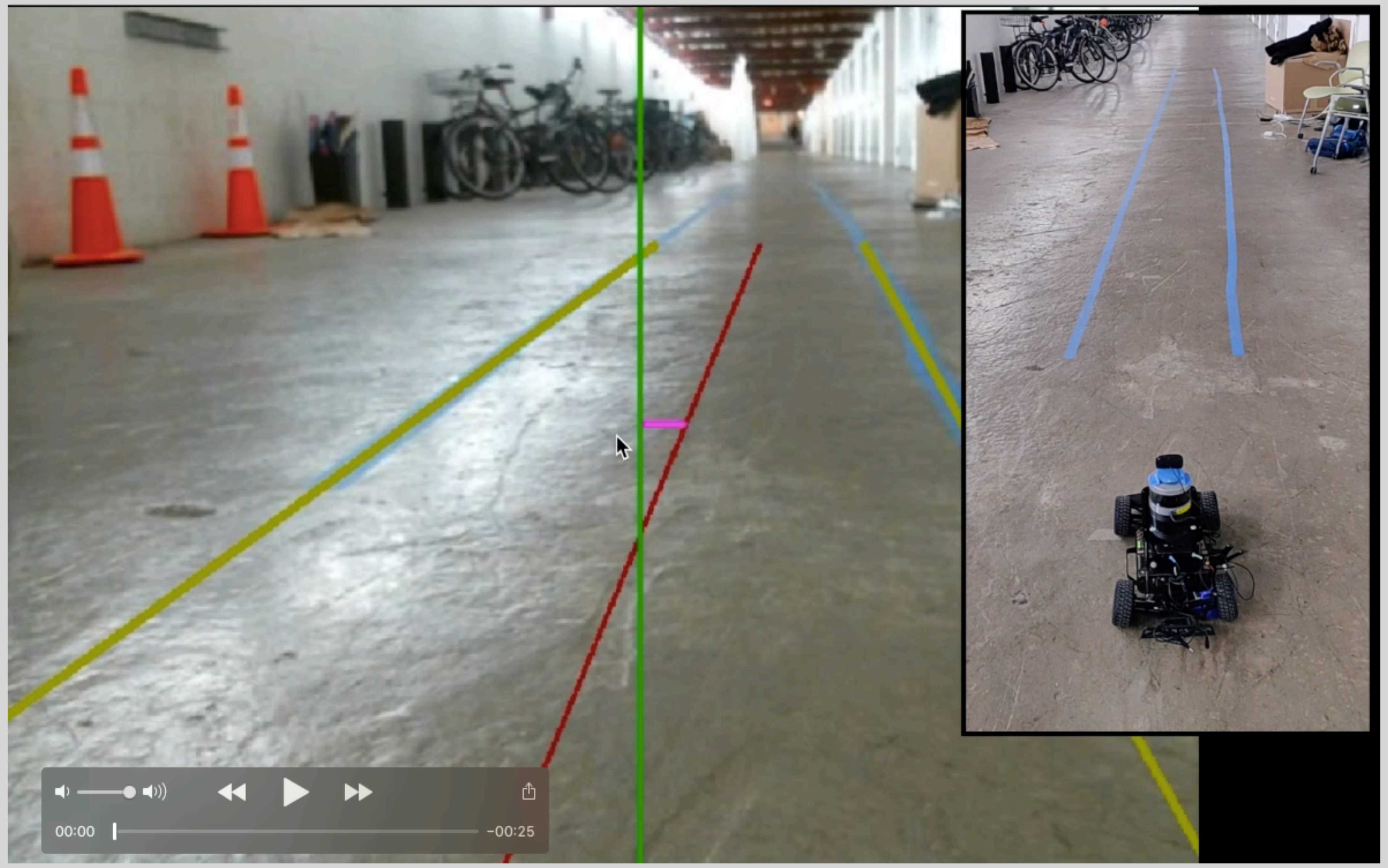


standard color filter: gets both lane lines wrong

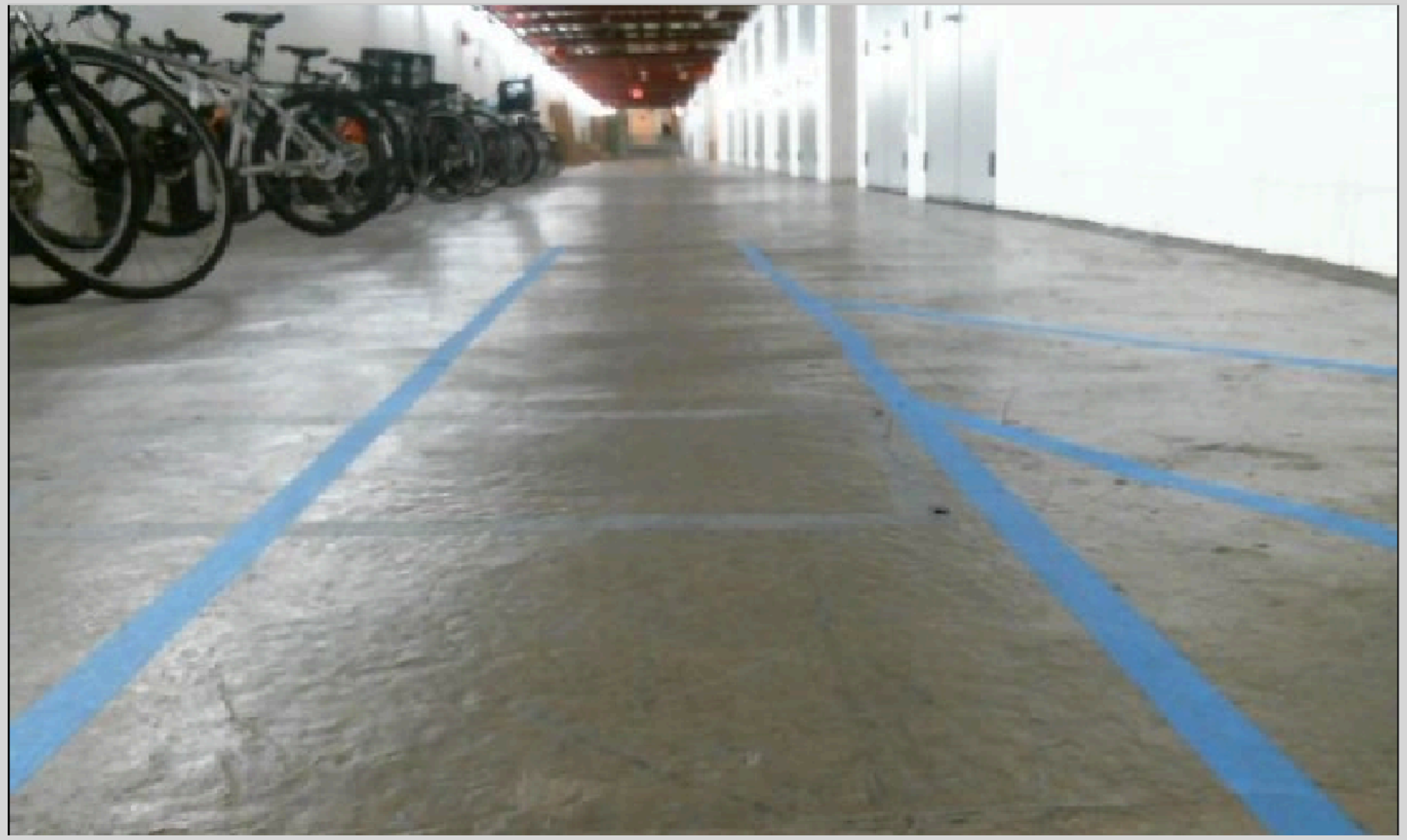


revised color filter: gets both lane lines right

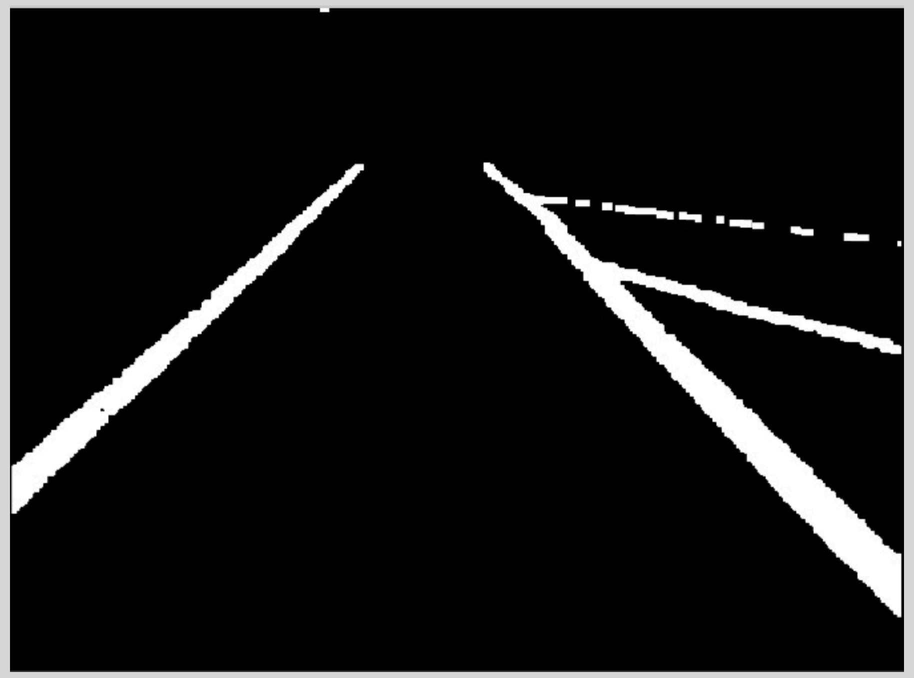
racecar experiment



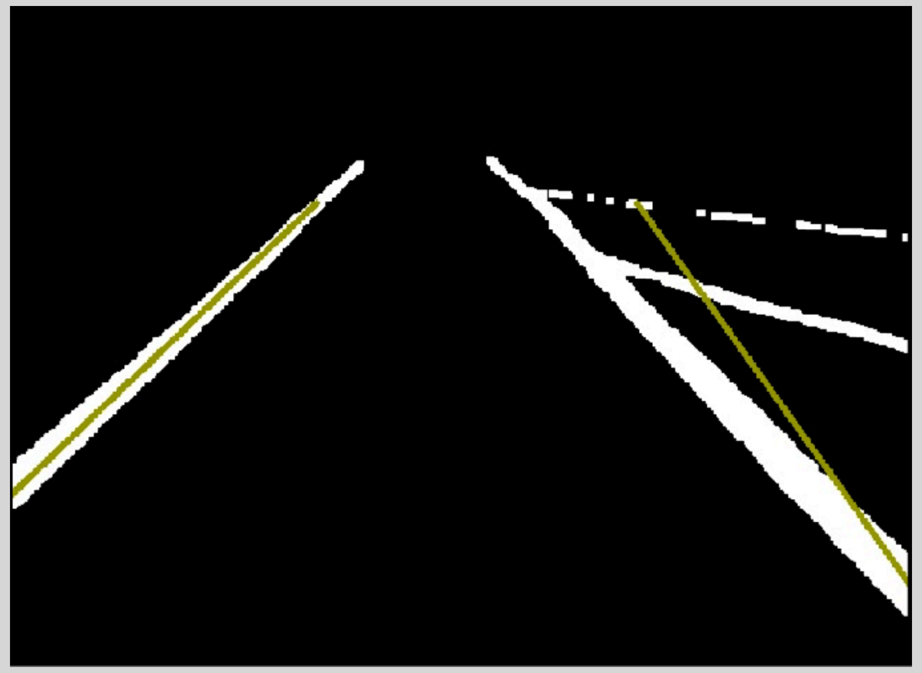
naive lane following algorithm



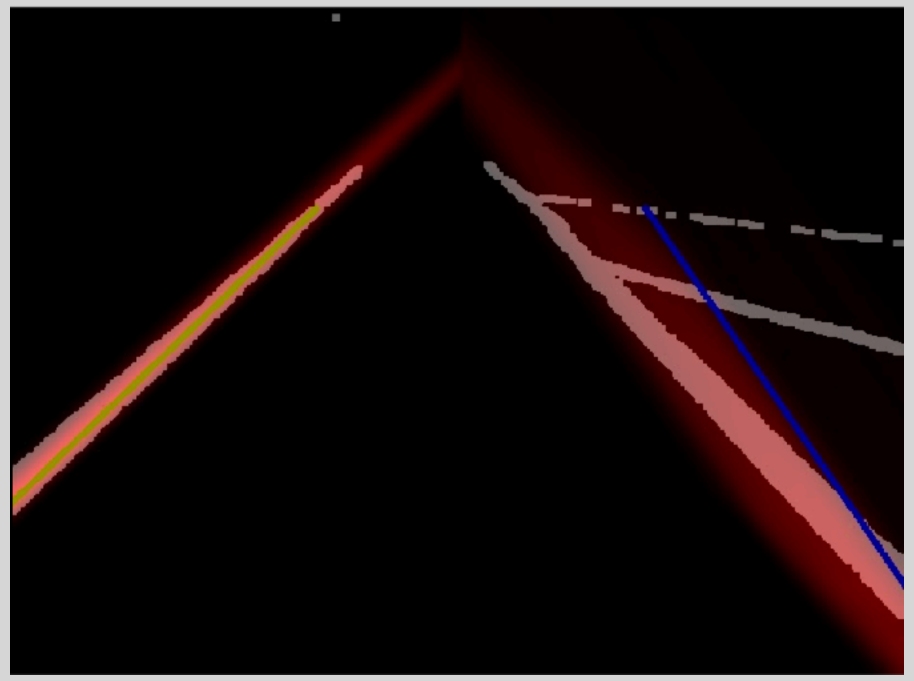
extra tape added on right to confuse controller



segmentation results



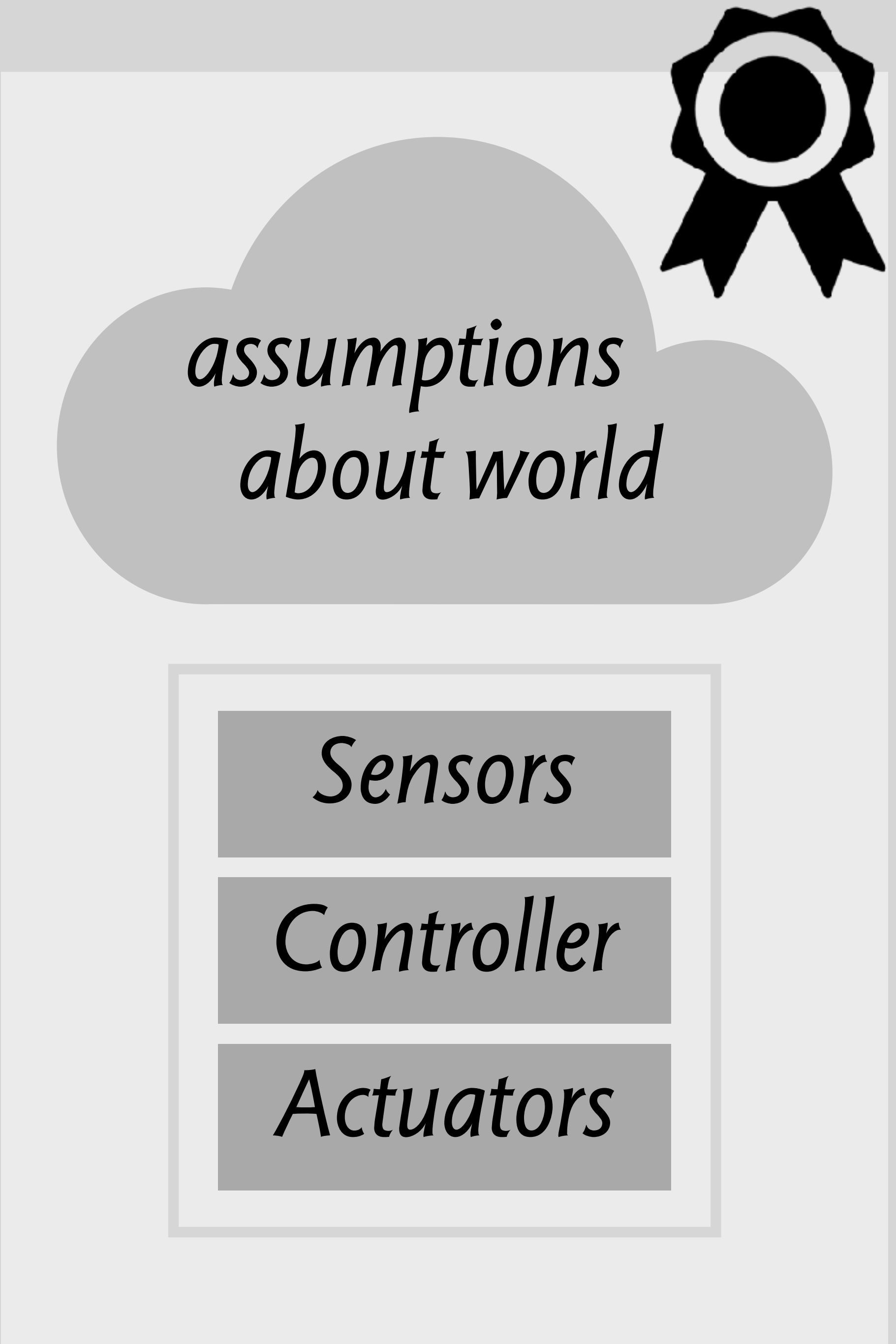
inferred lane lines



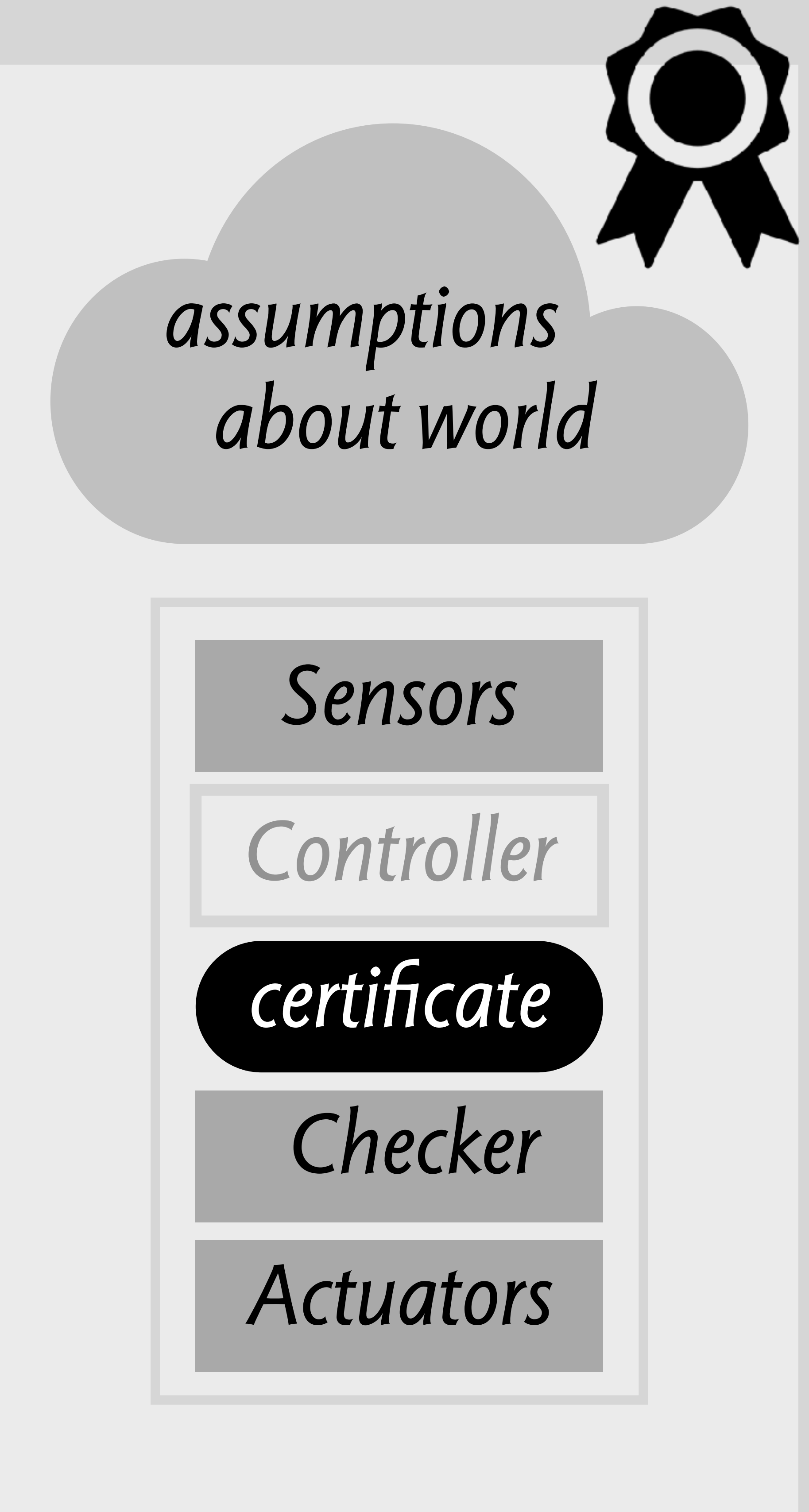
convolution result: reject

conclusion

assurance cases



assurance case for
traditional system



assurance case for certified control
trusted base excludes controller

some distinctions

being safe vs. being confident

incidental safety is not enough
public will demand evidence

anomaly detection vs. assurance case

great work on anomalies in machine learning
consistency between frames, common sense
but assurance case goes further: an argument for safety

best effort vs. explicit safety

today's controllers try to do their best
no explicit articulation of what's achieved
certificate articulates design consensus
eg: LiDAR point density sets size of smallest obstacle

next steps

simulations and trials

end to end simulation in racecar

integration with Toyota algorithms

testing in variety of conditions

design issues

certificate designs for different risks

formal verification of safety case